



Titre: Étude empirique du travail collaboratif ad hoc réalisé au cours d'un
Title: processus de génie logiciel

Auteur: Sébastien Cherry
Author:

Date: 2009

Type: Mémoire ou thèse / Dissertation or Thesis

Référence: Cherry, S. (2009). Étude empirique du travail collaboratif ad hoc réalisé au cours
Citation: d'un processus de génie logiciel [Thèse de doctorat, École Polytechnique de
Montréal]. PolyPublie. <https://publications.polymtl.ca/144/>

 **Document en libre accès dans PolyPublie**
Open Access document in PolyPublie

URL de PolyPublie: <https://publications.polymtl.ca/144/>
PolyPublie URL:

**Directeurs de
recherche:** Pierre N. Robillard
Advisors:

Programme: Génie informatique
Program:

UNIVERSITÉ DE MONTRÉAL

ÉTUDE EMPIRIQUE DU TRAVAIL COLLABORATIF *AD HOC*
RÉALISÉ AU COURS D'UN PROCESSUS DE GÉNIE LOGICIEL

SÉBASTIEN CHERRY
DÉPARTEMENT DE GÉNIE INFORMATIQUE ET GÉNIE LOGICIEL
ÉCOLE POLYTECHNIQUE DE MONTRÉAL

THÈSE PRÉSENTÉE EN VUE DE L'OBTENTION
DU DIPLÔME DE PHILOSOPHIAE DOCTOR
(GÉNIE INFORMATIQUE)
SEPTEMBRE 2009

UNIVERSITÉ DE MONTRÉAL

ÉCOLE POLYTECHNIQUE DE MONTRÉAL

Cette thèse intitulée :

ÉTUDE EMPIRIQUE DU TRAVAIL COLLABORATIF *AD HOC*
RÉALISÉ AU COURS D'UN PROCESSUS DE GÉNIE LOGICIEL

présentée par : CHERRY Sébastien

en vue de l'obtention du diplôme de : Philosophiae Doctor

a été dûment acceptée par le jury d'examen constitué de :

M. DESMARAIS Michel, Ph.D., président

M. ROBILLARD Pierre-N., Ph.D., membre et directeur de recherche

M. ANTONIOL Giuliano, Ph.D., membre

M. LÉVESQUE Ghislain, Ph.D., membre

REMERCIEMENTS

J'aimerais d'abord remercier sincèrement mon directeur de recherche, Monsieur Pierre N. Robillard, dont la vision, les conseils et les commentaires partagés au cours des longues heures de discussion passées à son bureau ont définitivement contribués à l'achèvement de ce travail.

Aussi, j'aimerais remercier de tout cœur l'organisation où les données ont été récoltées, sa directrice, pour m'avoir gracieusement autorisé d'investir l'endroit et surtout, j'aimerais exprimer ma plus sincère gratitude au gestionnaire de projet ainsi que tous les membres de l'équipe observée qui ont généreusement accepté de participer à cette étude et sans qui celle-ci aurait été tout simplement impossible. À tous ces gens que je ne peux nommer, mille mercis.

J'aimerais également remercier les membres du jury pour leurs commentaires constructifs lors de la présentation de cette recherche dans le cadre de l'examen synthèse de doctorat, les réviseurs anonymes lors de la soumission des différents articles soumis dans le cadre de cette recherche, ainsi que certains anonymes de l'assistance lors des conférences données pour leurs questions et commentaires pertinents.

J'aimerais grandement remercier Jean-François Morin pour son support lors de la rédaction de cette thèse. Merci également à Michel Martin pour sa précieuse contribution lors de la mise en place des outils de collecte de données de même que lors de la période de validation des données amassées.

J'aimerais finalement remercier chaleureusement ma complice de tous les jours, Caroline Chouinard, pour son support indéfectible tout au long de la rédaction de cette thèse et enfin, je ne saurais comment remercier ma mère pour m'avoir insufflé le goût du dépassement et sans qui tout ceci n'aurait été possible.

RÉSUMÉ

Considérée d'abord et avant tout comme une tâche purement technique, la construction de logiciels est néanmoins de plus en plus reconnue comme une activité définitivement humaine. Tandis qu'elle constitue une tâche cognitive intense en plus de représenter «un exercice en complexes interrelations» tel qu'évoqué par Fred Brooks en 1975 (Brooks, 1975), plusieurs facteurs humains sont de ce fait impliqués lors du développement d'un logiciel. Ce document présente donc une recherche empirique qui a été entreprise dans le cadre d'une étude de cas industrielle et qui a pour but d'analyser l'un des nombreux aspects humains impliqués en génie logiciel, soit le travail collaboratif *ad hoc* qui a lieu entre les concepteurs de logiciels au cours d'un processus de génie logiciel.

Les motivations pour l'entreprise de cette recherche sont nombreuses, entre autres, parce que certaines recherches telles que celle conduite par Perry, Staudenmayer et Votta (Perry, Staudenmayer, & Votta, 1994-) aussi bien que celle menée par Robillard et Robillard (Robillard & Robillard, 2000) ont indiqué l'importance insoupçonnée de ce phénomène en terme de temps octroyé par les concepteurs au cours d'un projet logiciel. En dépit de cette importante réalité, aucune étude connue n'a tenté de caractériser ce type d'activités ainsi que les communications qui en découlent afin d'en déterminer l'impact sur le processus de génie logiciel dans son ensemble. Par conséquent, il était permis de croire qu'une meilleure compréhension de ce type d'activités aiderait à mieux comprendre plusieurs aspects centraux du génie logiciel tels que la collaboration, la coordination et, d'un point de vue plus général, la dynamique au sein d'une

équipe de concepteurs de logiciels en plus d'aider ultimement à la proposition de certaines améliorations aux processus de génie logiciel dans le but de rendre la collaboration plus efficace entre les différents intervenants impliqués lors d'un développement de logiciel. En outre, cette meilleure compréhension pourrait également aider à la définition d'environnements de développement de logiciels plus adaptés comportant des outils de communication servant à supporter et faciliter la collaboration au sein d'une équipe autant localisée au même endroit que distribuée géographiquement, puisqu'il a été constaté que la distance, dans ce dernier contexte, lève des barrières aux communications informelles, comme celles étudiées au cours de cette recherche, conduisant à d'importants problèmes de coordination (Herbsleb & Grinter, 1999).

Afin d'attaquer cette ambitieuse entreprise, une méthodologie de recherches adaptée pour l'étude de l'objet spécifique d'intérêt aussi bien que le terrain sur lequel se déroule la recherche a été conçue et exécutée. Cette méthodologie, en grande partie inspirée des sciences humaines telles que la sociologie et la psychologie, fut composée d'une observation participante où l'observateur prit le rôle de participant complet en tant que membre à part entière de l'équipe observée. Pendant la longue période d'observations qui dura plusieurs mois, diverses méthodes de collecte de données ont été employées, incluant l'enregistrement audio-vidéo de sessions de travail, la capture des courriers électroniques échangés par les membres de l'équipe ainsi qu'une sauvegarde du code source du logiciel développé de même que tous les documents trouvés sur le serveur de fichiers partagé par l'équipe.

L'analyse de la grande quantité de données amassées pendant la phase d'observations a été réalisée par le biais de la technique issue de l'*analyse séquentielle et exploratoire de données*, connu sous le nom anglais de *Exploratory Sequential Data Analysis* (ESDA) telle que proposée par Fisher et Sanderson (Sanderson & Fisher, 1994; Fisher & Sanderson, 1993-). Cette technique particulièrement indiquée pour les recherches exploratoires et descriptives telles que celle-ci propose, entre autres, la transformation des données qualitatives en données quantitatives afin de réduire la variabilité des données et faciliter de subséquents calculs statistiques pour obtenir un éventail de données dérivées additionnelles.

À la suite de la longue phase d'analyse réalisée à partir de cette technique, plusieurs résultats ont été obtenus, résultats qui confirment d'abord l'importance du phénomène, mais qui révèlent de façon plus importante la nature *opportuniste* des activités collaboratives *ad hoc* observées et le rôle des différents acteurs identifiés qui y prennent part. Les résultats obtenus et leurs impacts ont été ainsi divulgués et discutés par le biais de plusieurs communications scientifiques tant dans le cadre de plusieurs conférences internationales que par l'entremise d'articles de revue. Trois de ces articles forment d'ailleurs le corps du présent document.

Finalement, des résultats obtenus de la présente étude, plusieurs hypothèses peuvent être induites et validées par d'éventuelles recherches confirmatoires forgeant de ce fait une nouvelle base théorique de connaissances dans le domaine. Cette meilleure compréhension de ce phénomène, qui était jusqu'à nos

jours pratiquement inexploré, pourra également guider la formalisation de nouvelles pratiques visant, tel qu'évoqué ci-dessus, à rendre la collaboration au sein d'une équipe de concepteurs de logiciels plus efficace. D'ailleurs, certaines pratiques comme telles ont également été proposées au cours des articles soumis et publiés.

ABSTRACT

Recognized above all as a technical task, the construction of software is nevertheless more and more acknowledged as a genuine human experience for the most part. While being an intensive cognitive task in the same time as it also constitutes “an exercise in complex interrelationships” as expressed by Fred Brooks in 1975 (Brooks, 1975), many human factors are thus involved during the development of a software. This document presents an empirical research that was undertaken in a industrial software engineering setting with the aim of examining one of the various human aspects of software development: the *ad hoc* collaborative activities taking place between teammates during a software development process.

The motivations for this endeavour were numerous, but among others, because some researches such the one conducted by Perry, Staudenmayer and Votta (Perry et al., 1994-) as well as another piloted by Robillard and Robillard (Robillard & Robillard, 2000) revealed the unsuspected importance of this phenomenon in term of time spent by the developers along an entire software project. Despite this really important reality, no known research has tried to characterize this type of activities as well as the communication that ensue in order to determine their impact on the whole software process. Therefore, it was legitimate to believe that a better understanding of this type of activities would help to better comprehend many pivotal aspects of software engineering, such as collaboration, coordination and, more generally, work team dynamics and help us to propose ultimately further process enhancements with the intend of making

collaboration during a software engineering process more efficient between involved stakeholders. Furthermore, this better comprehension could help as well the definition of more adapted software development environment comprising tools intended to support collaboration in a team, as much localized in the same office as geographically distributed, since that it has been found that the distance, in the latter context, raises barriers to informal communications, such as those studied by this research, leading to exaggerated coordination problems (Herbsleb & Grinter, 1999).

To tackle this ambitious enterprise, a research methodology adapted for the study of the particular object of interest as well as the study field setting had to be designed and executed. This methodology, largely inspired from human sciences such as sociology as well as psychology, consisted of a participant observation where the observer took the role of a complete participant as a full-fledged member of the team. During the fairly extensive observation period that lasted several months, various data collection methods have been employed and included the audio-video recording of working sessions, the capture of electronic mails exchanged by the team members as well as a back up of the source code of the software developed by the observed team and all artefacts found on the file server shared by the team.

The analysis of the mass of data collected during the observation phase has been realized by the technique of the Exploratory Sequential Data Analysis (ESDA) as proposed by Fisher and Sanderson (Sanderson & Fisher, 1994; Fisher & Sanderson, 1993-). This technique particularly well-suited for exploratory and

descriptive researches as this one proposes, among others, the transformation of qualitative data into quantitative data in order to reduce the variability of the data and to facilitate further statistical calculations to obtain an array of additional derived data.

From the quite extended analysis phase that have been realized following this technique, several interesting results were obtained, results confirming firstly the importance of the phenomenon but more importantly the *opportunistic* nature of *ad hoc* collaborative activities and the role of different identified actors who are taking part. Several obtained results and their impacts have been revealed and discussed in the course of several scientific communications such as conferences and journal papers. Three of these articles constitute the body of the present document

Finally, from the obtained results of the present study, a number of hypothesis could be inducted and validated by further confirmatory researches forging a new theoretical base of knowledge in the domain. This better understanding of the phenomenon, which until today was practically unexplored, can also guide subsequent practices formalization in order to attempt, as stated above, a number of such practices were also proposed in the course of submitted and published papers.

TABLE DES MATIÈRES

REMERCIEMENTS.....	III
RÉSUMÉ.....	V
ABSTRACT	IX
TABLE DES MATIÈRES	XII
LISTE DES TABLEAUX	XVII
LISTE DES FIGURES.....	XVIII
LISTE DES SIGLES ET ABRÉVIATIONS	XXI
LISTE DES ÉCHANTILLONS	XXIII
LISTE DES ANNEXES.....	XXIV
AVANT-PROPOS.....	XXVI
INTRODUCTION.....	1
CHAPITRE 1 REVUE CRITIQUE DE LITTÉRATURE	9
1.1 L'approche par processus en génie logiciel	9
1.1.1 L'apparition de la notion de processus en génie en logiciel.....	9
1.1.2 La mesure et l'amélioration continue des processus.....	13
1.2 Le travail collaboratif en génie logiciel	21
1.2.1 Les principes du travail d'équipe	21
1.2.2 L'importance du travail collaboratif.....	28
1.3 La psychologie de la communication.....	35
1.3.1 La communication au sein des groupes.....	35

1.4 Le développement distribué de logiciels	40
1.4.1 Les motivations d'affaires	41
1.4.2 Les problèmes de communication rencontrés.....	42
1.4.3 Les solutions envisagées	43
1.5 Les méthodes empiriques en génie logiciel	44
1.6 Sommaire de la littérature	46
CHAPITRE 2 PROTOCOLE DE RECHERCHE	48
2.1 Énoncé du problème	48
2.1.1 Objectifs généraux	50
2.1.2 Pertinence théorique de la recherche.....	53
2.1.3 Pertinence pratique de la recherche	54
2.2 Stratégie de recherche.....	55
2.2.1 Approche générale.....	56
2.2.2 Population cible.....	61
2.2.3 Échantillon et contexte de l'étude	61
2.2.4 Méthodes de collecte des données	64
2.2.5 Méthodes d'analyse des données	73
2.2.6 Considérations de validité et fiabilité de la recherche	99
CHAPITRE 3 PRÉSENTATION DES ARTICLES.....	114
CHAPITRE 4 THE SOCIAL SIDE OF SOFTWARE ENGINEERING – A REAL	
AD HOC COLLABORATION NETWORK.....	119
4.1 Abstract.....	119
4.2 Introduction	120
4.3 Research Methodology	122

4.3.1 Methodological Traditions	124
4.3.2 Observer or Participant.....	125
4.3.3 Industrial Environment Observed	127
4.3.4 Observation Phases	128
4.3.5 Data Recording	129
4.3.6 Ethical Issues.....	130
4.3.7 Sampling and coding.....	130
4.4 Role Definition	133
4.5 Measurement of <i>Ad hoc</i> Collaborative Activities.....	134
4.6 Concluding Remarks	143
4.7 Acknowledgements.....	147

CHAPITRE 5 AD HOC COLLABORATION IN A SOFTWARE DEVELOPMENT

SETTING – WHY AND HOW?.....	148
5.1 Abstract.....	148
5.2 Introduction	149
5.3 Methodology.....	151
5.3.1 Data Modeling.....	153
5.4 Channels of <i>Ad Hoc</i> Collaboration	155
5.4.1 Role Definition	157
5.5 Types of <i>Ad Hoc</i> Collaborations	160
5.6 Discussion and Concluding Remarks.....	164
5.6.1 Major findings.....	164
5.6.2 Validity Concerns.....	166
5.7 Acknowledgments	167

CHAPITRE 6 EMPIRICAL ANALYSIS OF <i>AD HOC</i> COLLABORATION	
PATTERNS AND CONTENT IN A CO-LOCATED SOFTWARE	
DEVELOPMENT ENVIRONMENT – A CASE STUDY	168
6.1 Abstract.....	168
6.2 Introduction	169
6.3 Methodology.....	172
6.3.1 Data Modelization.....	175
6.3.2 Validity Concerns.....	177
6.4 Channels of <i>Ad hoc</i> Collaboration.....	179
6.4.1 Usage of Communication Channels.....	180
6.4.2 Communication Channel Preferences	181
6.4.3 Duration of Communication Channels	184
6.4.4 Occurrence of Communication Channels	186
6.4.5 Participants of Communication Channels	187
6.5 Types and Content of <i>Ad hoc</i> Interactions	188
6.5.1 Types of Vocal <i>Ad hoc</i> Sequences	189
6.5.2 Types of Electronic <i>Ad hoc</i> Communications	194
6.5.3 Content (what) of <i>Ad hoc</i> Sequences.....	196
6.5.4 Focus of <i>Ad hoc</i> Communications	199
6.5.5 Relevance Level of <i>Ad hoc</i> Sequences.....	200
6.6 Discussion.....	206
6.6.1 Opportunistic Nature of <i>Ad hoc</i> Collaborations	207
6.6.2 Task-centric Content of <i>Ad hoc</i> Communications	209
6.6.3 Reserved Use of E-Mails in Co-Located Settings	211
6.6.4 Research Concerns	211

6.7 Conclusion	214
6.8 Acknowledgments	215
CHAPITRE 7 SOMMAIRE DES CONTRIBUTIONS	216
7.1 Apports méthodologiques	216
7.2 Apports théoriques	218
7.3 Apports à la pratique	220
CONCLUSION.....	223
RÉFÉRENCES	226
ANNEXES.....	239

LISTE DES TABLEAUX

Tableau 2.1 Résultats de la mesure d'accord intra et inter-codeurs en fonction des métriques retenues	108
Tableau 2.2 Résultats de la mesure d'accord inter-codeurs en fonction des métriques retenues et pour les activités identifiées par les deux codeurs	112
Table 5.1 <i>Ad hoc</i> collaboration channel occurrences per participant	159
Table 5.2. Frequency distribution of <i>ad hoc</i> collaboration types per channel	161
Table 5.3 Frequency distribution of vocal <i>ad hoc</i> collaboration type per participant	162
Table 6.1 Intra- and inter-coder agreement results	178
Tableau C.1 Liste des enregistrements audio-vidéos exclus pour accoutumer les sujets à la présence de la caméra	244
Tableau C.2 Liste des enregistrements audio-vidéos utilisés pour fins d'analyse ...	246

LISTE DES FIGURES

Figure 1.1 Grille de gestion de Blake et Mouton	26
Figure 2.1 Durée d'enregistrement et échantillonnage des sessions de travail	82
Figure 2.2 Processus d'analyse à l'aide de l'ESDA.....	85
Figure 2.3 Diagramme orienté-objet du modèle de données	89
Figure 4.1 Recording duration and sampling per working session day and period ..	132
Figure 4.2 Percentage of time spent on collaborative activities per subject for each recorded session	135
Figure 4.3 Average percentages of <i>ad hoc</i> collaborative work observed per subject.....	136
Figure 4.4 Average frequencies of <i>ad hoc</i> collaborative interactions per hour by subject.....	137
Figure 4.5 Cumulative numbers of <i>ad hoc</i> collaborative interactions as a function of their duration	138
Figure 4.6 Percentage of interactions initiated by the subject.....	139
Figure 4.7 Percentage of interaction involvement per subject	140
Figure 4.8 Network of <i>ad hoc</i> collaboration between the subjects and other stakeholders	141
Figure 4.9 Ratio of <i>ad hoc</i> collaborations that contribute to knowledge-sharing	143
Figure 5.1 The characteristics of <i>ad hoc</i> communications	153
Figure 5.2 Data model designed for encoding the raw data.....	154
Figure 5.3 Frequency distribution from the Face-to-Face, Verbal, and E-mail <i>ad hoc</i> collaboration channels	156
Figure 6.1 Data model designed for encoding the raw data.....	176

Figure 6.2 Percentage of time spent on <i>ad hoc</i> collaboration per communication channel.....	180
Figure 6.3 Overall communication channels and mode usage during <i>ad hoc</i> collaboration	181
Figure 6.4 Interchangeable <i>ad hoc</i> communication mode usage	183
Figure 6.5 <i>Ad hoc</i> communication channel mode patterns	184
Figure 6.6 Log-normal probability density distribution of vocal <i>ad hoc</i> collaborations as function of their interval duration	186
Figure 6.7 Distribution of <i>ad hoc</i> communication channels as a function of the number of participants involved	187
Figure 6.8 Frequency distribution of types of <i>ad hoc</i> sequences realized by the vocal communication channel	190
Figure 6.9 Duration distribution of types of <i>ad hoc</i> sequences realized by the vocal communication channel	190
Figure 6.10 Median duration of types of <i>ad hoc</i> sequences realized by the vocal communication channel	192
Figure 6.11 Frequency distribution of types of <i>ad hoc</i> sequences realized by the electronic communication channel.....	194
Figure 6.12 Frequency distribution of types of <i>ad hoc</i> sequences with respect to the communication channels	195
Figure 6.13 Frequency distribution of <i>ad hoc</i> sequence content by means of the vocal communication channel	197
Figure 6.14 Distribution of content as a function of the type of vocal <i>ad hoc</i> sequence.....	198

Figure 6.15 Percentage of <i>ad hoc</i> collaborations per communication channel as a function of the number of sequences per interaction	199
Figure 6.16 Frequency distribution of vocal <i>ad hoc</i> collaboration relevance levels .	201
Figure 6.17 Frequency distribution of relevance levels as a function of the type of vocal <i>ad hoc</i> sequences	203
Figure 6.18 Frequency distribution of relevance types as a function of vocal <i>ad hoc</i> communication modes.....	205
Figure B.1 Plan de l'espace de travail de l'environnement observé et emplacement de la caméra et des microphones	242
Figure D.1 Outil d'analyse exploratoire et séquentielle de données – outil de saisie de données	250
Figure D.2 Outil d'analyse exploratoire et séquentielle de données – outil de saisie de données et écran de saisie	250
Figure D.3 Outil d'analyse exploratoire et séquentielle de données – diagramme gant des sessions de travail analysées	251
Figure D.4 Outil d'analyse exploratoire et séquentielle de données – diagramme gant des interactions observées lors d'une session de travail analysée et détails d'une interaction	251
Figure D.5 Outil d'analyse exploratoire et séquentielle de données – rapport d'analyse des sessions de travail analysées	252

LISTE DES SIGLES ET ABRÉVIATIONS

Abréviation	Définition
CMM	Capability Maturity Model®
CMMI	Capability Maturity Model® Integration
ESDA	Exploratory Sequential Data Analysis
FAQ	Frequently asked questions
FtF	Face-to-face (face à face); désigne une interaction collaborative <i>ad hoc</i> qui implique un déplacement de la part de l'un ou l'autre des participants afin que ceux-ci puissent échanger verbalement face à face.
F2F	Désigne une interaction collaborative <i>ad hoc</i> face-à-face (FtF) impliquant strictement deux participants.
FnF	Désigne une interaction collaborative <i>ad hoc</i> face-à-face (FtF) impliquant au moins trois participants.
ISO	International Organization for Standardization
OTAN	Organisation du Traité de l'Atlantique Nord
PMBOK	Project Management Body of Knowledge
SEI	Software Engineering Institute
SWEBOK	Software Engineering Body of Knowledge
UML	Unified Modeling Language

UP	Unified Process
UPEDU	Unified Process for EDUcation
W3C	World Wide Web Consortium
XML	Extensible Markup Language
XP	eXtreme Programming

LISTE DES ÉCHANTILLONS

Échantillon 1 Exemple type de données utilisées pour l'analyse.....	98
Échantillon 2 Exemple type étendu de données utilisées pour l'analyse	254

LISTE DES ANNEXES

Annexe A Certificat d'acceptation du projet de recherche par le Comité d'éthique de la recherche avec des sujets humains de l'École Polytechnique de Montréal	239
Annexe B Plan de l'espace de travail de l'environnement observé et emplacement de la caméra et des microphones	241
Annexe C Liste des enregistrements audio-vidéos et échantillonnages réalisés pour l'analyse	243
Annexe D Images écran de l'outil conçu pour l'analyse exploratoire et séquentielle des données	249
Annexe E Échantillon type de données utilisées pour l'analyse	253
Annexe F Schéma XML utilisé pour la sérialisation et validation des données encodées sur support physique.....	259
Annexe G Audio Video Recording of <i>Ad hoc</i> Software Development Team Interactions Proceedings of CHASE 2009 - International Workshop on Cooperative and Human Aspects of Software Engineering, 31 st International Conference on Software Engineering Vancouver, Canada, 2009.....	267
Annexe H Importance of Peer-to-Peer <i>Ad hoc</i> Interaction in the Development of Large Software Systems <i>Actes de l'Ergo'IA 2006 - L'humain comme facteur de performance des systèmes complexes</i> Bidart-Biarritz, France, 2006	277
Annexe I Communication Problems in Global Software Development: Spotlight on a New Field of Investigation <i>Proceedings of The Third</i>	

<i>International Workshop on Global Software Development (GSD 2004)</i>	
Édimbourg, Écosse, 2004.....	285
Annexe J Empirical Study of <i>Ad hoc</i> Collaborative Activities in Software	
Engineering Proceedings of CSAC 2004 - First International Workshop	
on Computer Supported Activity Coordination Porto, Portugal, 2004.....	291

AVANT-PROPOS

« To improve the rigor of measurement in software engineering, we need not restrict the type or range of measurements we can make. Indeed, measuring the unmeasurable should improve our understanding of particular entities and attributes, making software engineering as powerful as other engineering disciplines. Even when it is not clear how we might measure an attribute, the act of proposing such measures will open a debate that leads to greater understanding. »

Norman E. Fenton et Shari Lawrence Pfleeger, auteurs du livre *Software Metrics : A Rigorous & Practical Approach*

INTRODUCTION

Le génie logiciel, plus jeune des disciplines du génie, fit son apparition il y a maintenant plusieurs décennies en réponse à la crise du logiciel, expression qui fut désignée à l'époque et dès lors couramment employée dans la littérature pour faire état du nombre très élevé de ratées dans la conduite de projets logiciels (Naur & Randell, 1969).

Depuis, la discipline fit des pas de géant par l'introduction de divers processus de contrôle et de gestion de projets ainsi que de diverses mesures et de métriques afin de pouvoir tendre vers un meilleur contrôle de la qualité des logiciels développés. C'est aussi dans ce même élan que la notion de processus de génie logiciel fit son apparition, principe qui vise à concentrer les efforts à déployer sur le processus duquel le produit logiciel est obtenu plutôt que sur le produit logiciel même. L'apport de cette nouvelle approche est qu'elle permet d'augmenter les chances de succès de projets logiciels en améliorant la prédictibilité des échéanciers et des coûts en plus de favoriser la production d'un logiciel de plus grande qualité (Robillard, Kruchten, & d'Astous, 2003).

Pourtant, encore de nos jours, certains problèmes persistent toujours et plusieurs auteurs en viennent à ce constat, auteurs tels que Brooks dans son article intitulé « No Silver Bullet » (Brooks, 1987-), Gibbs dans « Software's Chronic Crisis » (Wayt Gibbs, 1994) et Pressman lorsqu'il parle de « software's chronic affliction » (Pressman, 2001). Or, de plus en plus d'efforts sont déployés par plusieurs entreprises productrices de logiciels pour usages internes ou pour fins commerciales

dans une optique de faire de l'amélioration continue de leur processus de génie logiciel, certaines mettant même sur pied des unités vouées entièrement à cette fin. C'est dans ce contexte que s'inscrit la recherche proposée dans le présent document.

D'autre part, il est maintenant largement supporté dans la littérature que multiples problèmes rencontrés au cours de processus de développement de logiciels ne sont pas nécessairement d'ordre technologique, mais peuvent être également imputables à certains facteurs humains impliqués au cours d'un processus de génie logiciel. De ce fait, certains aspects tels que la communication (Herbsleb & Mockus, 2003-; Seaman & Basili, 1997), la coordination (Grinter, Herbsleb, & Perry, 1999; Herbsleb & Grinter, 1999), de même que la collaboration (Robillard & Robillard, 2000; Robillard, d'Astous, Detienne, & Visser, 1998) retiennent l'attention de plus en plus de chercheurs. Toutefois, alors que ce nouveau domaine de recherches gagne sans cesse en intérêt parmi la communauté scientifique, de nouveaux défis méthodologiques émergent. En effet, les facteurs humains du génie logiciel ont été longuement outrepassés pendant bon nombre d'années et ce, pour diverses raisons, mais principalement pour la difficulté inhérente à la mesure quantitative des aspects humains impliqués au cours d'un développement de logiciel (Perry et al., 1994-). Toutefois, les études empiriques réalisées en génie logiciel depuis les dernières années tendent à se raffiner de plus en plus afin d'étudier ce nouveau sujet d'intérêt, l'homme derrière la machine, usant désormais de plusieurs méthodes et techniques qui étaient autrefois réservées aux sciences humaines telles que la psychologie, la sociologie et l'anthropologie.

Plus particulièrement, en raison des énormes dépassements de temps observés lors de nombreux projets logiciels et dans le but de comprendre un peu mieux de quelle façon les concepteurs de logiciels emploient leur temps au cours d'un processus de développement de logiciel, Perry, Staudenmayer et Votta (Perry et al., 1994-) ont entrepris une étude exploratoire au cours de laquelle ils observaient le travail réalisé par les concepteurs. Ils ont entre autres observé que plus de 75 minutes par jour, par concepteur, étaient employées sous forme de communications informelles, que ce soit sous formes de conversations face à face, en conversations téléphoniques, par échanges de courriels, etc. Ces mêmes observations furent également corroborées par Robillard et Robillard (Robillard & Robillard, 2000) au cours d'une autre étude de cas lors de laquelle ils ont dénoté que plus de 41% du temps des concepteurs de logiciels était employé sous forme de ce qu'ils ont désigné «activités collaboratives *ad hoc*». Cela représente donc une part relativement importante du temps total alloué à un projet logiciel et occupée de façon informelle en dépit de la nature formelle des activités prescrites par les processus de génie logiciel. Cependant, aucune recherche connue à ce jour n'a tenté de décrire le contenu de ce type d'activités.

D'autre part, en raison du fait qu'une large part du travail d'un concepteur de logiciels consiste à recueillir une vaste quantité d'information afin de pouvoir les cristalliser en un logiciel qui répondra réellement aux besoins de l'utilisateur, et ce, dans le temps et les coûts prévus tout en rencontrant les critères de qualité exigés, il est maintenant largement accepté que ce processus ne pourrait se faire de façon efficace sans une collaboration toute aussi efficace entre les membres de l'équipe (Robillard et al., 2003). Il était donc permis de croire qu'une meilleure

compréhension des activités collaboratives *ad hoc* telles qu'observées par Robillard et Robillard (Robillard & Robillard, 2000), de même que de leur contenu et la nature des communications informelles qui en découle, aiderait par la suite à proposer certaines améliorations aux processus de développement de logiciels, améliorations qui seraient plus adaptées à la réalité empirique du développement de logiciels.

De plus, une autre motivation qui porte à vouloir comprendre ce type de travail collaboratif est issue du domaine de développement distribué de logiciels où il a été démontré que la distance dans un tel contexte dresse des barrières aux communications informelles, ce qui résulte en divers problèmes de coordination (Herbsleb & Grinter, 1999). Cette problématique est d'autant plus d'actualité, étant donné que pour bon nombre de raisons d'affaires, cette pratique est de plus en plus adoptée dans l'industrie (Herbsleb & Moitra, 2001).

Ce document présente donc un projet de recherche empirique qui eut cours dans le cadre d'une étude de cas en profondeur dans l'industrie et qui explore l'aspect du travail collaboratif en génie logiciel, mais plus spécifiquement les activités collaboratives *ad hoc* retrouvées lors d'un processus de génie logiciel. Tout comme Robillard et Robillard (Robillard & Robillard, 2000), par «activités collaboratives *ad hoc*» il est entendu les activités qui ne sont pas prescrites formellement, qui ont lieu entre deux ou plusieurs concepteurs travaillant ensemble sur une tâche particulière du projet et qui surviennent de façon informelle et spontanée. Elles peuvent prendre diverses formes telles qu'en conversations face à face, en échanges par courriers électroniques, en appels téléphoniques, etc.

Ainsi, cette recherche de nature exploratoire, c'est-à-dire qu'elle ne consiste pas à valider empiriquement des hypothèses formulées d'avance, mais prend plutôt fondement d'une question de recherche, propose une approche afin d'entreprendre la mesure et l'analyse du travail collaboratif *ad hoc* retrouvé au cours d'un processus de génie logiciel, puisque tel qu'il fut évoqué précédemment, aucune recherche connue à ce jour n'a tenté de réaliser une telle étude sur le sujet. Il fallut donc développer d'une part, la stratégie de recherche pour ce faire, et d'une autre part, la représentation formelle des données utilisée pour l'encodage de celles-ci. La stratégie de recherche est pour sa part constituée d'une collecte de donnée réalisée par le biais d'une observation participante telle que décrite par Jorgensen et Babbie (Jorgensen, 1989; Babbie, 2001), de même que l'analyse de données largement basée sur les fondements de l'*Exploratory Sequential Data Analysis* (ESDA), protocole d'analyse utilisé dans divers domaines lorsque l'intégrité séquentielle des données est essentielle pour être menée à bien. Quant à la représentation formelle des données formée à partir des observations directes sur le terrain, elle fut dérivée et illustrée à l'aide du formalisme combiné de la théorie de la linguistique et du paradigme orienté-objet inspiré par d'Astous et Robillard (Robillard et al., 1998; d'Astous & Robillard, 2000; d'Astous, 1999) et constitue la pierre d'assise des analyses qui y découlent.

Suite à une validation du schéma de codage de données utilisé par une approche intra et inter-codeurs afin de démontrer la validité de celui-ci, une analyse de la vaste quantité de données collectées et encodées est effectuée. S'appuyant à la fois sur les données quantitatives et qualitatives amassées, l'analyse est réalisée sous deux angles d'approches. D'abord l'analyse de contingence permet de comprendre

et d'éclaircir les caractéristiques événementielles du phénomène étudié. Ensuite, l'analyse de contenu caractérise enfin le contenu de ces activités, tant au niveau des sujets discutés que des types d'activités collaboratives *ad hoc* réalisées et identifiées à l'aide du schéma de codage précédemment élaboré.

Le chapitre 1 de ce travail constitue donc une revue de la littérature pertinente et du bagage théorique essentiel que le lecteur se doit d'avoir en tête avant de poursuivre dans le cœur de l'ouvrage. Cette revue débute par un rappel de la notion de processus, comment elle est apparue, pourquoi elle est primordiale en génie logiciel et quelles sont les principales approches de mesure et d'amélioration de processus. Ensuite, un rappel des notions entourant le travail d'équipe est établi, suivi des quelques résultats de recherche pertinents sur le travail collaboratif et certains éléments de littérature récents qui effleurent le sujet, de même qu'une revue des principaux fondements théoriques de la psychologie de la communication et de la communication au sein des groupes. Ensuite, quelques notions touchant le développement distribué de logiciels sont évoquées pour finalement terminer par un rappel de la place de la méthode empirique en génie logiciel.

Le chapitre 2 expose le protocole de recherche mis en place pour réaliser le présent projet de recherche. Cela inclut d'abord l'énoncé du problème d'intérêt de même que les objectifs poursuivis, suivi de l'approche générale de recherche, les différentes méthodes de collecte de données et d'analyses employées ainsi que les diverses considérations de validité et de fiabilité à observer.

Le chapitre 3 introduit de façon chronologique les trois articles de revue rédigés et qui forment le corps de la présente thèse par articles. Ainsi le chapitre 4 constitue le premier des trois articles qui a pour but de présenter le projet de recherche d'un point de vue sociologique, vu son objet d'étude, en y incluant les détails méthodologiques pertinents et certains résultats qui ont été obtenus par le fruit d'une analyse quant aux rôles occupés par les acteurs prenant part aux activités collaboratives *ad hoc* observées. Le chapitre 5 présente le deuxième article qui a pour but, quant à lui, de fournir les premiers éléments de réponses à savoir *pourquoi* et *comment* sont réalisés les activités collaboratives *ad hoc*. Une analyse du point de vue des différents acteurs introduits lors de l'article précédent, est également exposée. Quant au troisième article présenté au chapitre 6, il a pour but d'exposer des résultats d'analyses plus approfondis quant aux patrons et au contenu analysés des activités collaboratives observées rendant possible une meilleure compréhension de la dynamique des activités collaborative *ad hoc* au sein d'une équipe de concepteurs de logiciels et permettant l'élaboration de pratiques ayant comme objectif d'améliorer la collaboration dans un environnement de développement de logiciels.

Par la suite, le chapitre 7 dresse un sommaire des principaux apports de cette recherche à la science du génie logiciel, apports qui ont trait tant au point de vue méthodologique de recherches portant sur ce type sujet, qu'à la modélisation et l'amélioration de la pratique du génie logiciel.

Finalement, la conclusion du présent travail apporte en ouverture certaines pistes de travaux de recherche futurs pertinents qui pourraient être entrepris en

s'appuyant sur la base des résultats obtenus dans le cadre de cette étude et qui permettraient également de faire avancer l'état des connaissances en génie logiciel un peu plus loin.

En outre, les annexes jointes à ce document contiennent plusieurs informations pertinentes telles qu'un échantillon type de données ayant servi à l'analyse, des images-écrans de l'outil réalisé pour ce faire, mais inclus également l'intégralité de quatre articles rédigés et présentés dans le cadre de conférences internationales et qui ont permis tant la présentation de la recherche que de porter l'attention sur le phénomène étudié, mais également la divulgation de résultats préliminaires pour lesquels il a été possible de recueillir l'appréciation et les commentaires de la communauté de chercheurs qui s'intéressent également aux aspects humains du génie logiciel.

CHAPITRE 1

REVUE CRITIQUE DE LITTÉRATURE

La littérature sur le sujet spécifique de cette recherche qu'est l'étude du travail collaboratif en génie logiciel n'est pas très volumineuse, voire quasi-inexistante pour ce qui est précisément du travail collaboratif *ad hoc*. Cela démontre donc l'originalité du projet de recherche présenté dans ce document. De ce fait, la présente section visera à exposer quelques éléments de littérature pertinents et les assises théoriques desquels découleront la question de recherche formulée par la suite, le protocole de recherche permettant de répondre à cette question et finalement l'analyse des données subséquentes.

1.1 L'approche par processus en génie logiciel

1.1.1 L'apparition de la notion de processus en génie en logiciel

Depuis l'invention par Konrad Zuse, à la fin des années trente et au début des années quarante, du premier ordinateur programmable (Ritchie, 1986), la science de l'informatique n'a cessé d'évoluer à une vitesse effrénée. Les multiples usages de cette invention qui allait désormais bouleverser le monde moderne, et matérialisés sous forme de programmes informatiques, se sont accrus à une vitesse tout aussi fulgurante que les capacités de la machine physique elle-même.

Cette frénésie ne fit que s'amplifier exponentiellement avec le temps et très rapidement, dès la fin des années soixante, la taille et la complexité des programmes développés, dès lors désignés sous le nom de «logiciel», étaient devenues telles que l'on dû faire le constat d'une «crise du logiciel.» Cette expression, désignée pour la première fois en 1968 lors de la réunion d'un comité scientifique commanditée par l'Organisation du Traité de l'Atlantique Nord (OTAN) (Naur & Randell, 1969), est désormais communément utilisée dans la littérature pour faire état du nombre très élevé de ratées dans la conduite de projets logiciels de l'époque.

Par ailleurs, c'est à partir de ces années que naquit la science du «génie logiciel», discipline qui vint combler les nombreuses lacunes de la science informatique par l'instauration de processus, mesures et métriques afin de permettre aux ingénieurs de logiciels de contrôler toutes les phases et facettes d'un projet de développement de logiciel.

«Software engineering describes the collection of techniques that apply an engineering approach to the construction and support of software products. Software engineering activities include managing, costing, planning, modeling, analysing, specifying, designing, implementing, testing and maintaining. By "engineering approach," we mean that each activity is understood and controlled, so that there are few surprises as the software is specified, designed, built, and maintained. Whereas computer science provides the theoretical foundations for building software, software engineering focuses on implementing the software in a controlled and scientific way.» (Fenton & Pfleeger, 1996)

Depuis, la science du génie logiciel subit de nombreuses évolutions, elle qui n'en est certainement pas à ses dernières mutations. Parmi celles-ci, la notion de processus est probablement un des apports qui fit faire un bond de géant à la discipline. Ainsi,

contrairement à l'approche «produit» où tous les efforts sont employés à contrôler le produit à développer en tant que tel, l'approche «processus» préconise plutôt le contrôle rigoureux du processus duquel sera obtenu le produit logiciel à concevoir. Robillard, Kruchten et d'Astous (Robillard et al., 2003) définissent le processus de génie logiciel comme suit :

«The software process provides the guidelines for the efficient development and evolution of the software. It lays the groundwork for subsequent generations of the product; it reduces the possibility of risks and increases quality; and it captures and documents the best practices gleaned from previous experiences. A clearly defined software process is likely to promote a common vision and culture, while providing a road map for applying efficient tools. [...] The software process defines who is doing what, when, and how to reach a goal.»

De ce fait, un processus de génie logiciel consiste en la définition d'activités d'ingénierie et de gestion, conduites successivement ou en parallèle et menées en plusieurs phases, pour lesquelles seront requis un certain nombre d'intrants, tels que la définition des besoins, les connaissances techniques ou technologiques, et desquels en sortiront un certain nombre d'extrants sous forme de divers artefacts, tels qu'un document de spécifications, un document de design, le code source, les scripts de tests, etc. Finalement, le processus dans son ensemble doit mener en la cristallisation d'un ensemble d'information en un logiciel qui répondra aux besoins de l'utilisateur, dans les temps et les coûts prescrits, tout en rencontrant un certain nombre de critères de qualité (Robillard et al., 2003). De plus, comme l'ont souligné les auteurs précédemment cités, les processus de génie logiciel améliorent la prédictibilité des divers paramètres qui entrent en jeu lors du développement d'un logiciel en plus de permettre l'identification et la mitigation des risques liés au projet et augmenter, de surcroît, les chances de succès du projet logiciel. Enfin, de par

leur nature formelle et documentée, les processus de génie logiciel permettent également l'apport des expériences passées dans la définition et l'exécution des meilleures pratiques logicielles à conserver et perpétuent cette façon de faire pour les projets à venir.

Les avantages des processus de génie logiciel étant désormais clairement documentés et de plus en plus connus, une vaste gamme de processus a vu le jour dans la littérature aussi bien que sur le marché, chacun ayant leurs prémisses de base respectives sur lesquelles repose l'établissement de leurs grandes lignes directrices. On peut penser par exemple aux processus commerciaux tels que le bien connu Unified Process (UP) (Kruchten, 2000; Larman, 2002) ou encore aux méthodologies dites agiles, telles que l'eXtreme Programming (XP) (Beck, 2000) incluant également la famille des méthodologies «Crystal» (Cockburn, 2002), sans parler des processus sur mesure que les entreprises produisant des logiciels pour usage interne ou pour fins commerciales se sont développés pour leur propre usage.

Ainsi, cette approche «processus» gagne sans cesse en popularité et tend à se répandre, majoritairement dans les plus grandes entreprises ayant les ressources et le budget pour implanter de tels processus, certaines d'entre elles instaurant même des départements complets entièrement voués à l'amélioration continue de leur processus interne de développement de logiciels. Toutefois, même les plus petites organisations commencent à être de plus en plus sensibilisées aux nombreux avantages de cette approche. Cela ne doit certainement pas être étranger au fait que les notions si rattachant sont de plus en plus enseignées dans maintes universités de par le monde et dans le cadre de la plupart des cours de génie

logiciel dans lesquels les principes fondamentaux de l'approche «processus» sont inculqués et promus. Par exemple, le Unified Process for Education (UPEDU) (Robillard et al., 2003) se veut une version simplifiée et adaptée à l'enseignement du célèbre «Unified Process» de Rational (maintenant IBM) où l'apprenti ingénieur étudie les concepts fondamentaux et les différentes activités communes à tous bons processus de génie logiciel, et cela en établissant des liens directs et concrets avec l'un des processus les plus connus au monde, le «Unified Process».

1.1.2 La mesure et l'amélioration continue des processus

Comme le suggère la définition du génie logiciel citée précédemment et proposée par Fenton et Pfleeger (Fenton & Pfleeger, 1996), une approche d'ingénierie suppose que chacune des activités d'un quelconque processus est comprise et contrôlée. Or, comme le suggère DeMarco (DeMarco, 1982) dans sa célèbre proposition «you can't control what you can't measure», la mesure est une activité inhérente et essentielle en génie logiciel. Fenton et Pfleeger (Fenton & Pfleeger, 1996) décrivent la mesure comme suit : «Measurement is the process by which numbers or symbols are assigned to attributes of entities in the real world in such a way as to describe them according to clearly defined rules.»

Ainsi, la mesure permet de décrire des entités du monde réel en attribuant une valeur numérique ou symbolique aux attributs qui les constituent rendant possible, toujours selon les mêmes auteurs, la réalisation de trois types d'activités fondamentales. D'abord, elle aide à «comprendre» en rendant plus visible ou perceptible les différents aspects du processus et du produit et permettant ainsi une

meilleure compréhension des différentes activités et des entités affectées. Ensuite, la mesure permet de «contrôler» ce qui survient lors d'un projet en se servant d'une meilleure compréhension des entités et de leurs relations et en prévoyant ce qui est susceptible de se produire afin d'appliquer les changements requis pour que les buts du projet soient atteints. Finalement, toujours suite à une meilleure compréhension que procure la mesure, cette dernière permet en plus d'«améliorer» les processus et produits afin que les défaillances détectées ne se perpétuent pas aux projets subséquents.

De ce fait et comme il a été en partie évoqué ci-haut, la mesure en génie logiciel s'applique essentiellement à trois ensembles d'entités, soit aux entités reliées au produit, celles rattachées au processus et enfin celles relatives aux ressources. En ce qui a trait particulièrement aux entités touchant le produit, l'extrait suivant tiré d'un article intitulé «Making software development visible» de Hsia (Hsia, 1996) illustre assez bien la difficulté majeure et inhérente à la nature des logiciels, c'est-à-dire leur invisibilité.

«Software, like wind, is invisible yet powerful. We can 'see' wind only by observing its effect, as it sways branches and swirls sand. And we can 'see' software only by observing its functionality. Invisibility is what makes software development so difficult to manage -- it is hard to monitor the construction of something you can't see.»

Bien que la mesure du produit logiciel soit fondamentale en génie logiciel, elle est toutefois moins pertinente à la recherche exposée dans le présent document. D'autre part, concernant les mesures relatives aux entités liées au processus,

celles-ci bénéficient d'une attention accrue dans la littérature contrairement à ce dont elles avaient droit il n'y a encore pas si longtemps. Tel que discuté précédemment, de plus en plus de processus et méthodologies voient le jour et l'approche «processus» est maintenant largement adoptée au sein d'un nombre croissant d'organisations qui produisent du logiciel pour usage interne ou pour fins commerciales. Toutefois, l'approche «processus» en soit n'est pas une panacée. D'abord, un processus est loin de garantir la production d'un logiciel de qualité, mais de plus, tout bon processus n'est pas nécessairement applicable à toutes situations et dans toutes organisations. Non seulement faut-il être capable d'appliquer un processus adéquat à une organisation donnée dans un premier temps, mais encore faut-il être capable d'en mesurer son efficience pour être en mesure de l'optimiser dans un deuxième temps.

Ainsi, un exercice de mesures sur le processus de développement utilisé s'avère donc nécessaire en suivant une approche dite «bottom-up». Cette approche de mesure et d'amélioration de processus est celle préconisée par nombre d'auteurs dans la littérature. Contrairement à l'approche dite «top-down» qui consiste simplement à implanter un processus générique ou commercial ou qui consiste à valider le processus évalué par rapport à un modèle de référence, l'approche «bottom-up» consiste quant à elle à mesurer et à analyser le processus en place, déterminer les causes des problèmes rencontrés, pour ensuite en apporter les correctifs nécessaires (Briand, El Eman, & Melo, 1995; Robillard et al., 2003). C'est d'ailleurs cette approche qui sous-tend l'un des paradigmes de mesures de processus très bien connu sous le nom de «Goal-Question-Metrics» (Fenton & Pfleeger, 1996; Robillard et al., 2003).

Pour justifier l'adoption de l'approche «bottom-up», Briand, El Emam et Melo expliquent que le problème majeur rencontré à l'égard des processus génériques ou commerciaux sur lesquels on se base pour adopter une approche descendante (top-down), est qu'ils sont basés sur de nombreuses présomptions sur ce qui constitue de soit disant bonnes pratiques, présomptions qui ne sont pas vérifiées empiriquement et qui ne sont également pas toujours applicables dans certains contextes. De plus, les auteurs renchérissent en ajoutant que de nombreux problèmes qui sont rencontrés en pratique ne sont pas tous abordés par ces types de processus. L'approche ascendante (bottom-up) de l'amélioration du processus consiste donc, selon ces auteurs, à implanter une équipe d'amélioration du processus, modéliser le processus existant dans l'organisation concernée, mener une analyse qualitative afin d'identifier des sources de problèmes, définir et documenter un plan d'action ainsi qu'un programme de mesures, mener un projet pilote pour finalement implanter de nouvelles pratiques à l'ensemble de l'organisation.

Par ailleurs, on retrouve maintenant dans la littérature et sur le marché, un certain nombre de méthodes d'évaluation de processus s'inspirant également d'une approche «bottom-up». On n'a d'abord qu'à penser aux très célèbres «Capability Maturity Model[®]» (CMM) et «Capability Maturity Model[®] Integration» (CMMI) du Software Engineering Institute (SEI) (Paulk & Carnegie Mellon University. Software Engineering Institute, 1994; Ahern, Clouse, & Turner, 2001). Le CMM, lancé par le SEI en août 1990, connu dès lors un succès fort appréciable auprès des organisations en raison, entre autres, de sa présentation hiérarchisée des différents

niveaux de qualité. Plus spécifiquement, le CMM se veut d'être un modèle d'évaluation des processus organisationnels qui permet de classer le niveau de maturité du processus évalué en cinq niveaux. Ainsi, un processus classifié au niveau 1 indique un processus tout à fait *ad hoc* et chaotique où les chances de succès d'un projet sont tout à fait imprévisibles et dépendent grandement des individus impliqués. Le niveau 2, quant à lui, indique un début de gestion de projet, de spécifications, d'assurance qualité, etc., mais principalement pratiqués au niveau de chaque projet, alors que le niveau 3 indique des pratiques définies au niveau organisationnel. Le niveau 4 signifie la présence de mesures quantitatives de processus, alors que le niveau 5, le plus élevé du CMM, dénote finalement des activités d'optimisation de processus. De plus, les cinq niveaux de maturité définis par le CMM se subdivisent en 18 domaines clés, 52 buts et 316 pratiques clés donnant autant de points de repère spécifiques aux organisations désireuses de parfaire leur processus.

Une autre méthode d'évaluation de processus également populaire dans l'industrie est assurément celle de la norme ISO 9001 qui définit les standards d'assurance qualité relatifs à toutes les phases du cycle de développement du produit, soit de sa conception jusqu'à son installation et/ou sa mise en service (International Organization for Standardization (ISO), 2009). Cinq chapitres composent la norme, établissant les standards en terme de gestion de l'assurance qualité, des responsabilités de la haute direction, des ressources humaines, de l'ingénierie et construction des produits, ainsi que de la mesure, analyse et amélioration continue du processus. Un peu comme fonctionne le processus d'évaluation du CMM, le processus de certification ISO consiste en audits menés par des évaluateurs

externes qui recommanderont par la suite l'enregistrement de l'organisation évaluée ou non.

Ces deux types de méthodes d'évaluation de processus ne sont que quelques exemples bien connus parmi la manne de standards et de certifications qui existent sur le marché. Celles-ci comportent plusieurs avantages intéressants. Outre le fait qu'elles permettent l'adoption d'un processus susceptible de mieux épouser les formes procédurales organisationnelles, donc d'être mieux adapté au contexte spécifique dans lequel il est implanté, étant donné qu'elles consistent en l'établissement de standards sur ce qu'il doit être fait et en laissant les organisations libres d'en décider du comment, elles permettent également et justement de certifier de l'application d'un certain standard par l'organisation et de la mise en pratique de certaines règles de l'art bien établies dans le domaine. Aussi, non seulement de telles certifications permettent à l'organisation de dénicher de nouveaux contrats et d'attirer de nouveaux clients, elles sont, de plus, parfois une exigence stricte de plusieurs organisations qui pratiquent une gestion serrée de leurs fournisseurs et sous-traitants pour mener des affaires avec elles.

Toutefois, bien que cela puisse sans doute sembler surprenant et bien que ces types de certifications permettent d'attester que l'entreprise suit un processus standardisé à l'organisation, elles ne permettent pas toujours d'assurer que les produits, produits par l'entreprise, respectent un certain niveau de qualité. Autrement dit, une entreprise produisant des logiciels truffés d'anomalies, peut, par exemple, certainement obtenir une certification de type ISO 9001, en autant que son processus de production soit standardisé à l'organisation. Comme le soulignent

Fenton et Pfleeger dans leur livre intitulé *Software Metrics : A Rigorous & Practical Approach*, «quality, like beauty, is very much in the eyes of the beholder». Ainsi, l'important est de définir les attributs du logiciel présentant un intérêt pour l'utilisateur et de mesurer le degré de présence de ces attributs dans le produit logiciel (Fenton & Pfleeger, 1996). Toutefois, plusieurs écarts de non-qualité peuvent s'introduire, et ce, à plusieurs instants au cours du processus logiciel. Il peut y avoir, par exemple, un écart de non-qualité entre ce que l'utilisateur désire, et ce qu'il exprime, un écart de non-qualité entre ce que l'utilisateur exprime et ce que l'analyste comprend, un autre entre ce que l'analyste comprend et ce qu'il exprime par écrit dans le document de spécifications des requis logiciels, un autre encore entre ce qui est spécifié dans le document de spécifications et ce qui est compris par les concepteurs, un autre entre ce qui est compris et ce qui est réalisé par ces mêmes concepteurs, un autre entre ce qui est réalisé et ce qui est testé et finalement un autre entre ce qui est testé et ce qui aurait dû être testé. Même si tout bon processus de développement de logiciels doit théoriquement inclure les activités nécessaires afin de minimiser toutes ces écarts de non-qualité, plus souvent qu'autrement, la pratique indique que ce n'est malheureusement pas toujours le cas.

En fait, au niveau des processus mêmes, ils existent également ce que l'on pourrait appeler des «marges de non-conformité». D'abord, il existe une marge de non-conformité entre ce qui devrait être prescrit et ce qu'il l'est vraiment. Ensuite, la plupart des chercheurs seront d'accord pour dire qu'il existe une autre marge de non-conformité entre ce qui est prescrit et ce qui est réellement pratiqué par les concepteurs. Enfin, lorsque vient le temps d'être audité, les plus honnêtes

avoueront qu'il existe également, dans certains cas, un écart de non-conformité entre le processus réellement suivi par les concepteurs de logiciels et ce qui est permis de constater aux auditeurs. Plusieurs raisons peuvent expliquer ces marges de non-conformité, mais l'une des plus évidentes et la plus fréquemment soulevée par les praticiens du domaine, est que l'institutionnalisation des processus tel que promu par le CMM et l'ISO et leur simplification pour être applicable à l'ensemble des projets menés par l'organisation, provoque un effet pervers d'encourager l'adoption d'un processus souterrain beaucoup plus riche et adapté aux besoins spécifiques du contexte. «Institutionalization guarantees nothing, and efforts to institutionalize often lead to a bifurcation between an oversimplified public process and a rich private process that must be practiced undercover.» (Bach, 1994)

Finalement, quant au troisième et dernier type de mesures, celles concernant les entités reliées aux ressources, ce sont certainement celles ayant le moins été traitées dans la littérature, sans doute en raison de la difficulté inhérente à la mesure des facteurs humains influant au cours d'un processus de génie logiciel. Même si la mesure des ressources inclut aussi bien le matériel requis au processus de développement que les ressources humaines chargées de réaliser le projet, il n'en demeure pas moins que ces dernières constituent assurément l'élément le plus important et ayant le plus d'impact sur le projet. Ce type de mesures est par ailleurs le plus pertinent dans le cadre de la recherche présentée dans ce document et ne sera conséquemment pas traité dans le présent chapitre, mais sera abordé plus particulièrement au cours des chapitres qui suivent.

1.2 Le travail collaboratif en génie logiciel

1.2.1 Les principes du travail d'équipe

En génie logiciel comme dans nombre de domaines, aussi bien dans les sphères industrielles, sportives ou autres, même au sein du règne animal, les principes du travail d'équipe sont appliqués et la démonstration des prérogatives s'y rattachant n'est plus à faire. En fait, d'un point de vue macroscopique, l'organisation du monde moderne dans son ensemble constitue en soi un immense travail d'équipe, c'est-à-dire que d'une part, nous dépendons du fruit du travail des autres pour survivre et d'autre part, le travail d'équipe au sens large est inhérent à toutes formes d'organisations. C'est d'ailleurs depuis l'avènement de l'agriculture remontant aussi loin qu'à l'époque mésolithique, environ 9000 ans avant Jésus Christ, que l'on vit apparaître chez l'Homme certaines des formes organisationnelles plus structurées et qui se sont progressivement imposées jusqu'à nos jours (Angrignon & Ruelland G., 1995). Passant alors de la chasse à l'agriculture comme principale activité de subsistance, et ne pouvant ainsi plus assumer seul l'entièreté de ses besoins en raison du temps beaucoup plus important qu'exige la culture de la terre, dès l'époque néolithique, l'Homme a rapidement dû recourir à la division du travail en subdivisant et en spécialisant ses différentes activités sous forme de corps de métier. Les activités de l'un allaient désormais profiter à l'autre, et vice versa. Ainsi, on assista alors à la fois à une sédentarisation de l'Homme ainsi qu'à la formation des premiers villages dans lesquels les villageois avaient chacun leur rôle à jouer afin de subvenir aux besoins de la collectivité (Angrignon & Ruelland G., 1995). D'autre part, des milliers d'années plus tard, plus récemment lors de la révolution

industrielle, le concept de travail d'équipe subit une fois de plus des chambardements majeurs alors qu'Henry Ford mis au point la première chaîne de montage qui allait à tout jamais changer l'organisation du travail (Carlton & Perloff, 1994). L'ouvrier vit alors sa tâche se subdiviser à nouveau, devenant de ce fait «sur-spécialisé» dans l'accomplissement de cette dernière à un point tel qu'il en risquait même une forme d'aliénation au sens où elle fut brillamment illustrée par Charlie Chaplin dans son célèbre film «Les Temps Modernes» (Chaplin, Goddard, MK2 Productions, & Warner Home Video (Firm), 2003). Ces deux courts rappels historiques ne sont que des exemples servant à illustrer que la notion de travail d'équipe est loin d'être nouvelle et que son application peut être observée depuis des siècles aussi bien à un niveau macroscopique qu'à une échelle beaucoup plus réduite. Les pages qui suivent synthétiseront donc les principes généraux du travail d'équipe dans une optique appliquée plus particulièrement à la conception de logiciels.

D'entrée de jeu, l'omniprésence du travail d'équipe s'explique par sa raison d'être, celle d'obtenir ce qu'on appelle une synergie faisant en sorte que la quantité de travail produit par l'équipe soit supérieure à la somme des parties qui compose cette équipe. Ceci est attribuable principalement au fait que l'interaction et l'union des forces de chacun des équipiers amènent quelque chose de plus. Toutefois, l'obtention de cette synergie est loin d'être chose facile à réaliser et plusieurs facteurs peuvent influencer l'atteinte ou non de cette dernière.

D'abord, le travail d'équipe peut se manifester sous plusieurs formes. Comme l'expliquent Robillard, Kruchten et d'Astous (Robillard et al., 2003), une équipe peut

se composer tout simplement d'individus travaillant au même projet ou d'un groupe partageant les mêmes outils ou une même pièce. Toutefois, les «vraies équipes» au sens où l'entendent les auteurs, se composent d'individus qui possèdent des habiletés complémentaires partageant à la fois les mêmes buts, objectifs de performance et approches communes envers lesquels ils se tiennent mutuellement imputables. Ces quelques conditions ne constituent toutefois que la base des éléments requis à la formation d'équipes productives et efficaces.

En effet, Robillard, Kruchten et d'Astous (Robillard et al., 2003) énumèrent plusieurs prérequis à la formation d'équipes efficaces. Selon les auteurs, une équipe devrait être composée de plus ou moins cinq à sept membres consacrés au même projet. En premier lieu, les membres de l'équipe doivent impérativement tous consentir aux mêmes objectifs, desquels découlera un plan d'équipe qui définira le rôle de chacun de même que le processus de travail suivi par l'équipe. L'engagement des membres envers les objectifs, le rôle de chacun ainsi que le plan d'équipe est une condition de succès *sine qua non*. Ensuite, l'environnement de travail dans lequel évolue l'équipe doit également être propice au travail en équipe. De plus, une communication ouverte et libre pour tous, le respect mutuel ainsi que le support entre coéquipiers sont indispensables afin que les membres de l'équipe n'aient pas l'impression d'effectuer du travail en parallèle, mais aient vraiment le sentiment de réaliser un travail d'équipe.

Au plan organisationnel, la structure de l'équipe constitue également un des éléments importants quant aux chances de succès de l'équipe. Le choix de la structure doit se faire en fonction de divers facteurs incluant, mais ne se limitant

pas à : la durée du projet, le nombre de ressources affectées au projet, la période de disponibilité des ressources, le degré de spécialisation des ressources, le processus de génie logiciel suivi, etc. Toutefois, comme l'explique le corpus de connaissances en gestion de projet, le *Project Management Body of Knowledge*, le choix d'une structure d'équipe n'est pas rigide et celle-ci peut se caractériser, par exemple, d'un continuum allant d'une structure fonctionnelle, à une structure par projet (Project Management Institute, 2000). Ainsi, une structure fonctionnelle consiste en un regroupement de coéquipiers par spécialité et relevant d'un gestionnaire de spécialité au sein duquel il est possible de retrouver la notion de projet, mais où la portée du travail réalisé se limite aux frontières fonctionnelles assumées par le rôle de l'équipier. À l'opposée, la structure par projet se distingue par un regroupement d'équipiers relevant d'un gestionnaire de projet et assumant la majorité des fonctions requises à la bonne marche du projet. Entre ces deux formes structurelles se trouvent trois structures matricielles allant de faible à forte tout dépendamment du degré de caractéristiques de structure fonctionnelle ou par projet que l'on y retrouve (Project Management Institute, 2000).

De plus, plusieurs comportements d'équipe peuvent être adoptés tout dépendamment de la structure organisationnelle privilégiée. Pour expliquer les types de comportements qu'il est possible de recenser au sein des différentes équipes, Robillard, Kruchten et d'Astous (Robillard et al., 2003) proposent une analogie liée aux équipes sportives. Par exemple, dans de petites équipes telles qu'au tennis en double, chaque membre s'adapte à la position de l'autre afin de renforcer la solidité de l'équipe et couvrir dans ce cas-ci, le plus de terrain possible. Au sein de plus grandes équipes telles qu'au hockey, en dépit du fait que chaque

joueur occupe une position déterminée, l'ensemble des joueurs se déplace ensemble au gré du jeu sur l'ensemble de la patinoire. Enfin, le dernier comportement retrouvé est celui pratiqué par les équipes de baseball où tous les joueurs occupent une position fixe et sont tenus de la conserver pour ne pas affaiblir et découvrir certaines zones du terrain. Bien que ces comportements se distinguent tous grandement les uns des autres, à l'instar de la structure de l'équipe, il est difficile de statuer sur le comportement idéal à adopter par une équipe puisque le bon choix de celui-ci dépend d'un ensemble de facteurs propres à chaque situation. Par exemple, pour en revenir à l'analogie, il serait inopportun pour une équipe de baseball d'adopter le comportement d'une équipe de hockey et vice versa.

Par ailleurs, le style de gestion adopté par le gestionnaire de l'équipe revêt également d'une grande importance pour le bon rendement de l'équipe. Afin d'illustrer le large spectre en deux dimensions des différents styles de gestion qu'il est possible de recenser, Blake et Mouton (Blake Robert R. & Mouton Jane Srygley, 1964) ont constitué la grille suivante.

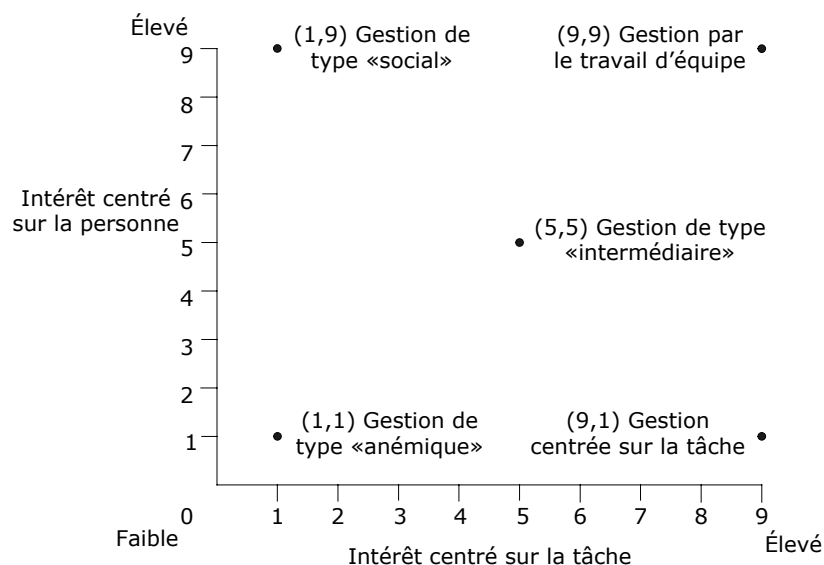


Figure 1.1 Grille de gestion de Blake et Mouton
Traduit et adapté de (Blake Robert R. & Mouton Jane Srygley, 1964)

Tel que le suggère la grille ci-dessus, le style de gestion (9, 1) centré sur la tâche est caractérisé par un leader qui considère le subordonné comme un outil de production et pour lequel l'essentiel du travail se résume *grosso modo* à planifier le travail dans les moindres détails, diriger, donner des ordres, contrôler les résultats et sanctionner. À l'opposé, le style de gestion (1, 9) centré sur la personne est reconnu lorsque le gestionnaire désire maintenir des relations harmonieuses dans le groupe sans se soucier de la production ou du rendement de ses subordonnés. Ce type de leader a horreur des conflits et cherche plutôt à plaire, à contenter tout le monde, sans exercer de contrôle et de sanction dans le but de se faire respecter. Le style de gestion (5, 5) de type «intermédiaire» constitue le compromis entre l'intérêt centré sur la tâche et celui centré sur la personne. Le gestionnaire s'y rattachant vise à un rendement adéquat sans plus, en cherchant à convaincre et motiver ses subordonnés et en faisant quelques compromis de temps à autres. Le style de gestion (1, 1) «anémique» quant à lui, est caractérisé par le gestionnaire

effacé et réfugié derrière les règlements, qui ne cherche qu'à passer inaperçu et qui ne se préoccupe ni de la tâche, ni des personnes qu'il dirige. Finalement, le style de gestion (9, 9) basé sur le travail d'équipe vise à un rendement élevé par la participation de tous en insistant sur la planification du travail, l'organisation et le contrôle, mais en cherchant à impliquer tout le monde dans le processus en déléguant les responsabilités. Ce style de leader ne cherche pas à éviter les conflits à tout prix, mais il les gère.

Bien que ces quelques notions de travail d'équipe semblent très pragmatiques, le processus de formation d'une équipe efficace est quant à lui beaucoup plus difficilement cernable. Tuckman (Tuckman, 1965), également référencé par Robillard, Kruchten et d'Astous (Robillard et al., 2003), propose un processus de développement d'équipe composé de quatre phases. La phase initiale consiste à la formation de l'équipe et est caractérisée par des relations où les membres de l'équipe affichent une conduite sociale appropriée, mais laissant place, par certains groupuscules, à certains jeux de coulisses lors desquels la direction peut être critiquée secrètement de même que certains autres membres de l'équipe. Cette phase généralement peu productive se poursuivra jusqu'au moment où les critiques et conflits seront discutés ouvertement par l'ensemble de l'équipe. Lors de la seconde phase, celle des conflits, certains désaccords et certaines oppositions émergent alors que les membres de l'équipe tracent les bornes de ce qu'ils sont prêts à accepter et à tolérer et argumentent par rapport aux rôles et aux tâches des uns et des autres. Certaines rivalités et concurrences peuvent également naître entre membres disputant le rôle de leader au sein de l'équipe. Bien que cette phase puisse être parfois douloureuse, elle est néanmoins un mal nécessaire à l'évolution

de l'équipe. C'est lors de la troisième phase, celle de la normalisation, que l'équipe commence à ressembler à une vraie équipe, où les conflits de toutes sortes laissent place à l'intérêt de l'équipe et où la culture et les normes de l'équipe sont développées. C'est lors de cette phase que l'équipe commence à produire efficacement. Finalement, la dernière phase, celle de la performance, est celle où l'équipe fonctionne à son meilleur, c'est-à-dire qu'elle devient amplement fonctionnelle, livre les meilleures performances et est également apte à répondre aux problèmes efficacement. Toutefois, pour atteindre ce niveau de maturité, chaque équipe se devra de passer à travers un processus de formation d'esprit d'équipe en cinq phases, tel que celui décrit par Robillard, Kruchten et d'Astous (Robillard et al., 2003), afin que les membres de celle-ci puissent pleinement travailler en coopération, maintenir une bonne performance et éviter de retomber en phase de formation.

1.2.2 L'importance du travail collaboratif

Tel qu'il fut évoqué en introduction, de plus en plus de chercheurs maintiennent que de nombreux problèmes survenant au cours de processus de génie logiciel ne sont pas tous attribuables à des lacunes d'ordre technique, mais peuvent être également imputables à des facteurs humains. Parmi ceux-ci, Perry, Staudenmayer et Votta (Perry et al., 1994-), également soutenus par Seaman (Seaman, 1999-), estiment que trop d'attention est portée sur des aspects technologiques et l'une des raisons fréquemment invoquées expliquent-ils, est la difficulté inhérente à la mesure quantitative de ces facteurs humains. Il est également permis de croire que la nature technique de la science du génie logiciel y est également pour quelque chose

étant donné que l'étude des aspects humains en génie logiciel implique inévitablement l'apport de sciences humaines, donc souvent hors des intérêts ou compétences techniques des chercheurs du domaine. Toutefois, un nombre grandissant de recherches portant sur divers aspects humains impliqués en génie logiciel ont été réalisées témoignant de l'importance de ce domaine de recherche.

De ce fait, l'un des aspects humains ayant retenu une certaine attention de la part de nombreux chercheurs gravite autour du champ de la communication entre les concepteurs de logiciels. Ainsi, certains chercheurs ont investigué précisément le phénomène de la communication au cours du développement de logiciels (Herbsleb & Mockus, 2003-; Seaman & Basili, 1997), alors que d'autres ont plutôt opté d'étudier les raisons sous-jacentes à ces communications qui sont, entre autres, l'aspect de la coordination (Grinter et al., 1999) ou d'une perspective plus large (Herbsleb & Grinter, 1999), l'aspect de la collaboration (Robillard et al., 1998; d'Astous & Robillard, 2000; Robillard & Robillard, 2000). Bien que ces trois aspects aient été généralement traités séparément, il est permis de penser qu'ils sont liés conceptuellement les uns les autres puisque la coordination peut être considérée comme étant un type parmi d'autres de collaboration et la communication comme étant la manifestation concrète d'une forme quelconque de collaboration.

Relativement à l'aspect de la collaboration, l'objet d'étude de la recherche présentée dans ce document, Robillard et Robillard (Robillard & Robillard, 2000) ont catégorisé empiriquement quatre types d'activités collaboratives au cours d'un processus de génie logiciel. Le premier, les activités collaboratives «obligatoires» sont définies comme étant les réunions d'équipe formelles et planifiées sur une base régulière,

alors que les activités dites «appelées» sont décrites comme étant les réunions planifiées par des membres d'une équipe pour résoudre un problème ponctuel et généralement technique. Les activités collaboratives «*ad hoc*» quant à elles regroupent le travail que les membres d'une équipe effectuent ensemble au même moment. Le dernier type d'activités collaboratives, les activités «individuelles», sont retrouvées lorsqu'un individu travaille seul sur une tâche du projet. Au cours de l'étude de cas réalisée par Robillard et Robillard (Robillard & Robillard, 2000), il a été recensé que 4% du temps passé au projet par les sujets observés était employé sous forme d'activités obligatoires alors que 14% du temps était occupé sous forme d'activités appelées. Enfin, les activités collaboratives *ad hoc* et les activités individuelles totalisaient respectivement 41% du temps passé au projet. Ainsi, l'une des conclusions apportées par cette dernière étude est à l'effet que les activités collaboratives *ad hoc* semblent jouer un rôle majeur dans la dynamique des communications d'équipe. En effet, comme l'étude l'a démontré, ce type d'activités occupe une part importante (41%) du temps alloué à un projet et constitue également les plus longues sessions de travail réalisées sur une base individuelle. Aussi, ces activités semblent avoir un impact important sur les activités individuelles puisqu'elles sont souvent réalisées par deux équipiers et précèdent souvent de longues sessions de travail individuel.

Les observations de l'étude de Robillard et Robillard (Robillard & Robillard, 2000) sont par ailleurs corroborées par Perry, Staudenmayer et Votta (Perry et al., 1994-) qui, quelques années auparavant, ont observé au cours d'une première étude de cas, que les communications informelles occupaient en moyenne approximativement 75 minutes par jour du temps des concepteurs de logiciels. Bien

que les chercheurs mentionnés ci-dessus aient quantifié le travail collaboratif *ad hoc* ou tout concept similaire, tel que celui des communications informelles, beaucoup d'autres chercheurs soulignent l'importance de ce type d'activités et expriment diverses opinions à cet effet.

D'un côté, Seaman et Basili (Seaman & Basili, 1997) soutiennent que ce type de communications revêt une importance significative et s'avère indispensable aux concepteurs de logiciels afin qu'ils puissent accomplir leurs tâches adéquatement. Karn et Cowling (J.S. Karn & A.J. Cowling, 2005) abondent dans ce sens et trouvent, à la suite d'une étude récente sur les effets de la perturbation sur les performances d'équipe de génie logiciel et l'impact des personnalités sur ces perturbations, que les équipes qui ne communiquent pas adéquatement causent plus de problème à la qualité du produit que les problèmes causés par les perturbations elles-mêmes. Chong et Siinon, également, considèrent les interruptions essentielles pour l'échange ou le gain d'informations requises pour l'obtention d'un produit de qualité (Chong & Siino, 2006). Enfin Ko, DeLine et Venolia (Ko, DeLine, & Venolia, 2007) indiquent que la sensibilisation ou la recherche d'information constitue d'importantes sources de communications informelles. Alors qu'ils définissent la première comme étant transmises afin de renseigner un collègue proactivement d'éléments d'information, la seconde est plutôt recherchée activement par celui qui les requière.

D'un autre côté, certains chercheurs discutent des aspects plus péjoratifs des communications informelles. Pour les évoquer, d'autres expressions pour désigner le même concept sont employées. Jett et George (Jett & George, 2003) définissent

d'abord le phénomène comme intrusions et soulèvent qu'elles forment une source d'interruptions lorsqu'une «rencontre inattendue est initiée par une autre personne qui interrompt le cours et la continuité du travail d'un individu et amène ce travail à un arrêt temporaire.» Les auteurs tempèrent en indiquant que même si les perturbations provoquées par les intrusions sont souvent considérées comme étant négatives d'une perspective de gestion des temps, celles-ci peuvent parfois être positives tout dépendamment le contexte dans lequel elles surviennent, par exemple lorsqu'elles impliquent un partage d'informations susceptible ne pas survenir par des moyens plus formels.

DeMarco et Lister (DeMarco & Lister, 1999) dans leur parution populaire intitulée *Peopleware : productive projects and teams*, considèrent également les communications informelles comme étant une forme d'interruptions et cadre celles-ci dans le fait qu'elles brisent ce qu'ils appellent le «mental flow», c'est-à-dire l'état mental et cognitif dans lequel se trouve un individu afféré à une tâche intellectuelle. Lorsqu'une interruption se produit, soutiennent les auteurs, la récupération de l'état mental dans lequel l'individu baignait avant que l'interruption se produise peut prendre jusqu'à 15 minutes impliquant une importante perte de temps due à ces interruptions étant donné leur occurrence significative survenant normalement dans un contexte de travaux d'équipe. D'ailleurs, plusieurs pratiques suggérées par les auteurs transpirent de cette vision un tant soit peu négative des interruptions notamment au chapitre 11 alors qu'ils énoncent certaines règles d'usage du téléphone ou au chapitre 12 *Bring Back the Door* du même ouvrage.

Pour leur part, Di Penta, Harman et al. (Di Penta, Harman, Antoniol, & Qureshi, 2007) dans une étude au cours de laquelle ils revisitent la célèbre loi de Brooks *Mythical Man Month* (Brooks, 1975) à l'effet que l'ajout de ressources à un projet augmente la surcharge de communication, explorent justement les effets de cette surcharge de communications sur les effectifs oeuvrant à des projets de maintenance de logiciels. Leur étude amène les chercheurs à la conclusion que les surcharges de communications constituent un aspect que les gestionnaires de projet doivent prendre en considération et indiquent qu'elles peuvent varier selon certains facteurs tels que la taille des équipes, le nombre d'équipes et plus globalement le nombre d'effectifs travaillant au projet.

D'un point de vue général, l'importance significative que revêt le travail collaboratif *ad hoc* ou tout concept s'y apparentant, au cours du processus de génie logiciel semble curieusement et de prime abord contraire à l'essence même de la notion de processus, c'est-à-dire celle d'organiser et de prescrire des activités de façon formelle dans le but d'en assurer le contrôle et la reproductibilité de même que de prescrire des artefacts nécessaires à la communication des éléments d'information essentiels au projet entre les divers intervenants impliqués. L'étude de Melnik et Maurer (Melnik & Maurer, 2004) visant à explorer le rôle de la conversation et des interactions "sociales" comme élément clé du partage de connaissance au sein des équipes de génie logiciel propose un élément de réponse à cette question. Leurs observations rapportent en effet que la documentation, en plus ne pas être facile à produire, constitue une forme de partage de connaissances écrite plus exigeantes à transmettre que verbalement étant donné que ce transfert est d'une part cognitivement plus rapide par la voie orale (Gould, 1978) et que celle-ci est d'autre

part moins exigeante (Kroll, 1978). Il semblerait d'ailleurs, que bien que la problématique de partage de connaissances soit particulièrement applicable au domaine du génie logiciel, celle-ci soit également transposable dans tout domaine où un important volume de connaissances doit être de la sorte partagé afin de mener à bien les activités à réaliser (Fischbach, Gloor & Schoder, 2009).

Plus récemment, alors que le rôle social du développement de logiciels est de plus en plus reconnu, certains chercheurs (Treude & Storey, 2009; Sarma, Maccherone, Wagstrom, & Herbsleb, 2009) tentent de lier les aspects sociaux et techniques impliqués en génie logiciel par l'instauration de pratiques et outils ayant pour objectifs d'améliorer ou de faciliter la collaboration entre les concepteurs de logiciels.

Toutefois, bien que les diverses études rapportées ci-dessus aient traitées d'une manière ou d'une autre et de façon plus ou moins spécifique du travail collaboratif *ad hoc* ou des communications informelles qui en découlent, de leur importance, de leurs impacts et des moyens d'améliorer ce type de collaboration dans un environnement de développement de logiciels par le biais de pratiques ou d'outils de collaboration adaptés, aucune étude trouvée n'a pourtant tentée d'étudier en profondeur et de décrire en détail les caractéristiques et surtout le contenu de ce type d'activités, connaissances, semble t'il indispensables à la bonne et juste compréhension de cette importante forme collaboration et préalable à la formulation d'approche à adopter pour l'adresser. Ce fait est donc à l'origine de la recherche décrite dans le présent document.

1.3 La psychologie de la communication

Comme il fut expliqué précédemment, le travail d'un concepteur de logiciels s'effectue principalement au sein d'une équipe de concepteurs et consiste essentiellement à cristalliser un ensemble d'information en un logiciel qui devra répondre aux besoins de l'utilisateur, dans les temps et les coûts prescrits (Robillard et al., 2003). Or, étant donné que d'une part, il est légitime de penser que cette cristallisation d'information n'aurait certainement pas lieu en l'absence de communication entre les membres de l'équipe et les divers intervenants qui y gravitent autour, et que d'une autre part, il est soutenu que pour obtenir une équipe mature et efficace, une bonne communication est nécessaire entre les membres de l'équipe, il est donc essentiel d'aller puiser quelques fondements théoriques dans le domaine de la psychologie de la communication et plus spécifiquement, de la communication au sein des groupes.

1.3.1 La communication au sein des groupes

Bien que la définition de ce qu'est un groupe puisse être très variable d'une référence à l'autre, selon Adler et Towne (Adler & Towne, 1999), un groupe se définit comme un «ensemble restreint de personnes qui interagissent, habituellement face à face, pendant une courte période de temps, dans le but d'atteindre des objectifs précis.» Ainsi, selon cette définition, pour que l'on puisse parler de groupe, il faut d'une part qu'il y ait interaction entre les personnes impliquées, il faut que cette interaction dure un certain temps, il faut également que cette interaction implique un certain nombre de personnes, soit entre trois et vingt

personnes, bien que cette dernière règle ne soit fixe, et enfin il faut que ces personnes partagent des objectifs communs à atteindre.

Comme Adler et Towne l'expliquent (Adler & Towne, 1999), les objectifs recherchés par les membres d'un groupe peuvent se subdiviser en deux groupes. On retrouve d'abord des objectifs individuels, incluant des objectifs liés à la tâche, c'est-à-dire qui concernent des tâches à accomplir sur le plan individuel ou encore des objectifs sociaux tels que des besoins d'appartenance, d'affection ou encore d'autorité. Ensuite, on retrouve des objectifs liés au groupe à proprement parler, c'est-à-dire le but recherché collectivement par les membres du groupe.

D'autre part, une fois le groupe constitué, il est pertinent d'en examiner les caractéristiques. D'abord, comme l'explique Babbie (Babbie, 1980), rapporté par Adler et Towne (Adler & Towne, 1999), le groupe est régi par certaines normes; ententes qui indiquent la façon dont les individus doivent se comporter les uns vis-à-vis les autres. Ainsi, les normes peuvent être distinguées en deux groupes, soit les normes dites «officielles», qui constituent des ententes définies publiquement et qui prescrivent explicitement les comportements appropriés de ceux qui ne le sont pas et enfin les normes dites «officieuses», qui ne sont pas définies ouvertement, mais qui ont autant de portées même si elles ne sont qu'implicites.

De plus, un groupe est également caractérisé par un certain nombre de rôles. Kowitz et Knutson (Kowitz & Knutson, 1980), rapportés par Adler et Towne (Adler & Towne, 1999), expliquent que bien que les normes définissent les critères acceptables par le groupe, les rôles constituent les schémas de comportements

attendus des membres du groupe. À l'instar des normes, il existe également des rôles reconnus officiellement, souvent accompagnés d'un titre de fonction tel qu'analyste, gestionnaire de projet ou architecte et on retrouve également d'autres rôles statués de façon plus officieuse, tels que celui du gourou, du farceur ou du délégué. De plus, certains rôles fonctionnels doivent être assumés par les membres du groupe afin que ce dernier puisse atteindre ses objectifs (Adler & Towne, 1999). Ces rôles se subdivisent en deux types, soit les rôles liés à la tâche et ceux liés à l'entretien des relations. Normalement implicites, ils sont habituellement établis et assumés par les membres du groupe sans qu'ils en soient conscients. Aussi, l'appropriation des rôles par les membres se fait en fonction de deux facteurs, soit en fonction des caractéristiques personnelles de chaque membre et en fonction de la particularité du groupe. Cependant, les deux problèmes majeurs rencontrés par rapport à l'attribution des rôles au sein d'un groupe se manifestent majoritairement lorsqu'un des rôles clés n'est pas rempli ou, à l'opposé, lorsque trop de membres se disputent le rôle en question. Dans les deux cas, le résultat est que la tâche liée au rôle n'est pas remplie, même lorsque trop de membres se disputent le rôle en raison des tensions internes générées par les aspirants.

Une autre caractéristique importante des groupes est le schéma d'interactions qui est établi (Adler & Towne, 1999) au sein du groupe. Ce schéma constitue le flot d'information entre les membres du groupe et est influencé par deux facteurs. Le premier est la disposition physique des individus au sein du groupe. En effet, certaines études, telles que celle de Steinzor (Steinzor, 1950), rapportée par Adler et Towne (Adler & Towne, 1999), démontrent par exemple que des personnes assises en cercle auront tendance à discuter avec les personnes placées en face

d'elle plutôt qu'avec celles assises à leurs côtés. Le deuxième facteur constitue quant à lui le réseau de communication établi entre les membres du groupe. Comme le rapportent Adler et Towne (Adler & Towne, 1999), la topologie du réseau sera fortement influencée par la situation du leader puisque celui-ci occupera souvent la position centrale au sein du réseau. Toutefois, dans le cas où il n'y a aucun leader officiel au sein du groupe, la personne placée au centre du réseau a plus de chance d'assumer le leadership (Shaw, 1981).

Par ailleurs, les dernières notions théoriques qu'il serait pertinent de revoir sont celles rattachées à la question du leadership au sein d'un groupe. Bien que l'on pense souvent au chef du groupe, le leader en titre, lorsque l'on pense au leadership, le rôle de leader n'est pas toujours assumé par un seul individu. En fait, comme l'expliquent Adler et Towne (Adler & Towne, 1999), le leadership est défini comme étant la capacité d'un individu d'influencer le comportement des autres membres du groupe et cette influence peut être exercée parce que le leader détient un ou plusieurs des pouvoirs suivants. On retrouve d'abord le pouvoir légitime, le pouvoir le plus en vue étant donné que la personne qui le détient l'a en raison de la position qu'elle occupe. On retrouve ensuite le pouvoir coercitif, qui est la capacité d'influer par le biais de menace et d'imposition de conséquences désagréables. Ce type de pouvoir est généralement exercé par le leader qui détient le pouvoir légitime, quoique ce n'est pas toujours le cas. Ensuite, on retrouve le pouvoir rémunérateur qui, à l'inverse du pouvoir coercitif, est la capacité d'influencer par la promesse ou l'octroi de conséquences agréables. Vient ensuite le pouvoir du spécialiste qui est détenu par celui à qui on reconnaît certaines compétences dans un domaine donné. Le pouvoir du référent est, quant à lui, détenu par un membre

du groupe à qui on voue une certaine forme de respect et sympathie. Enfin, le pouvoir de l'information est celui exercé par un membre qui détient certaines informations d'importances. Ainsi, tel que souligné ci-haut, le leadership n'est pas toujours assumé par le leader en titre et peut être partagé par plusieurs membres détenant diverses formes de pouvoir.

De plus, comme l'expliquent Cragan et Wright (Cragan & Wright, 1986) rapportés par Adler et Towne (Adler & Towne, 1999), les leaders affichent certaines caractéristiques distinctives. Il semblerait d'abord qu'ils possèdent une meilleure apparence physique, étant généralement plus grand, fort, physiquement plus beaux et ayant de meilleures aptitudes physiques. Aussi, ils seraient plus sociables que les autres, s'exprimant plus aisément et avec plus de facilité, étant plus enclins à coopérer facilement avec les autres et plus populaires que les autres. Les leaders possèderaient également certaines aptitudes qui aideraient les groupes à accomplir leurs tâches, telles qu'une grande intelligence et manifesteraient un plus grand sérieux. Finalement, les leaders seraient également investis du goût du leadership et feraient tout leur possible pour obtenir ce leadership en prenant souvent des initiatives, affichant beaucoup de persévérance et en exprimant souvent leurs convictions avec assurance.

Par ailleurs, trois formes de leadership sont pratiquées par les leaders à divers degrés (Adler & Towne, 1999). Le leadership autoritaire est pratiqué plus souvent qu'autrement par le leader détenant le pouvoir légitime, coercitif et rémunérateur ayant ainsi le pouvoir de dicter aux autres membres quoi faire. Ensuite, le leadership démocratique est pratiqué par le leader en titre qui laisse l'entière liberté

aux autres membres de participer aux prises de décisions. Enfin, le leadership du laisser-faire est pratiqué par le leader en titre qui renonce à son rôle officiel et qui laisse le groupe sans dirigeant. Ces trois dernières formes de leadership constituent également une autre forme de catégorisation du style de gestion des leaders à l'instar de la grille de gestion proposée par Blake et Mouton (Blake Robert R. & Mouton Jane Srygley, 1964) et présentée précédemment. Toutefois, comme les recherches de Fiedler (Fiedler, 1967) rapportées par Adler et Towne (Adler & Towne, 1999) le démontrent, tout style de gestion parmi ce large éventail n'est pas nécessairement bon à appliquer dans toutes situations et serait plutôt fonction de la relation entre les membres et le leader, du pouvoir détenu par le leader en titre de même que de la tâche à accomplir par le groupe.

1.4 Le développement distribué de logiciels

Dans le contexte actuel de mondialisation des marchés et de la main d'œuvre, les entreprises d'aujourd'hui n'ont désormais plus guère le choix d'user de tous les moyens à leur disposition afin de rester compétitives et maintenir leur part de marché. Le domaine du logiciel n'y fait malheureusement pas exception et c'est pour cette raison que le développement distribué de logiciels s'est vite imposé comme pratique incontournable pour tous les gros joueurs du domaine. Les sections qui suivent présenteront quelques-unes des motivations d'affaires qui justifient le recours à une telle pratique et les obstacles majeurs rencontrés par les organisations qui l'appliquent.

1.4.1 Les motivations d'affaires

De prime abord, le développement de logiciel distribué, également connu sous les expressions anglaises de «collaborative software development» ou encore «global software development», réfère au développement d'un logiciel de façon distribuée dans le temps et/ou dans l'espace, c'est-à-dire sur plusieurs sites et/ou sur plusieurs tranches horaires. Quoique le temps ne soit pas une composante nécessaire pour parler de développement distribué, l'espace quant à lui en est la prémisses de base. Toutefois, contrairement à ce qu'il est permis de penser, il ne faut pas nécessairement croire qu'une énorme distance doit séparer les équipes pour impliquer un travail distribué. En fait, selon Allen (Allen Thomas J, 1977), une distance de 30 mètres suffit pour observer une chute drastique des communications entre les membres d'une même équipe et ce, au même niveau que s'ils avaient été séparés de plusieurs kilomètres.

Ainsi, concernant les motivations d'affaires qui justifient le recours à une telle pratique, il n'est pas surprenant de constater que l'une des principales raisons évoquées est la réduction du coût de la main d'oeuvre et la disponibilité de cette main d'oeuvre en pays étrangers (Ebert & De Neve, 2001; Kobitzsch, Rombach, & Feldmann, 2001). Toutefois, Ebert et De Neve soulignent le fait que la distance entre les équipes amène un coût supplémentaire à la fois lié à la gestion des équipes virtuelles et à la moins bonne performance de ces équipes. D'autre part, une autre raison de l'accroissement de la pratique du développement distribué de logiciels serait attribuable aux nombreuses fusions et acquisitions des entreprises qui seraient ainsi forcées à conserver le capital de connaissances acquis (Grinter et

al., 1999; Herbsleb & Moitra, 2001). De plus, les avantages liés à la proximité du marché et des clients potentiels seraient également une autre source de motivation pour l'adoption d'une telle pratique (Ebert & De Neve, 2001; Mockus & Herbsleb, 2002). Finalement, la pression qui pousse à améliorer sans cesse le temps de mise en marché contribue également à inciter la pratique du développement de logiciel distribué puisque que la répartition de la force de travail sur plusieurs fuseaux horaires amène la possibilité de faire du développement «round-the-clock», c'est-à-dire sans interruption à raison de 24 heures par jour (Ebert & De Neve, 2001; Herbsleb & Moitra, 2001).

1.4.2 Les problèmes de communication rencontrés

Bien que la pratique du développement de logiciel distribué puisse amener son lot d'avantages, elle occasionne également un certain nombre de problèmes. Plus particulièrement, outre les problèmes d'ordres légaux, d'infrastructure (Kobitzsch et al., 2001), de culture et de langue (Kruchten, 2004) et ceux, à proprement parler, liés aux individus impliqués (Herbsleb & Mockus, 2003-; Herbsleb & Moitra, 2001), certains problèmes importants au plan de la communication peuvent également se manifester.

Comme il a été mentionné plus tôt, la communication constitue un ingrédient indispensable au développement de logiciels (Robillard & Robillard, 2000; Seaman & Basili, 1997; Herbsleb & Mockus, 2003-). Non seulement elle permet la cristallisation de l'ensemble des informations requises en un logiciel qui correspond aux besoins de l'utilisateur (Robillard et al., 2003), elle joue également un rôle majeur

à la bonne santé de l'équipe (Adler & Towne, 1999) ainsi qu'à sa coordination (Grinter et al., 1999; Herbsleb & Grinter, 1999). Toutefois, il a été trouvé, lors de recherches empiriques (Grinter et al., 1999; Herbsleb & Grinter, 1999) que dans un contexte de développement distribué, la distance entre les membres d'une équipe virtuelle semble lever certaines barrières aux communications informelles, celles qui surviennent à tout moment de façon spontanée, incluant également celles que certains appellent le «water-cooler talk» (Grinter et al., 1999). Les conséquences de cette baisse de communication sont multiples, mais entre autres, elles impliquent divers problèmes de coordination au sein de l'équipe (Herbsleb & Grinter, 1999), de gestion des connaissances et d'expertises (Herbsleb & Moitra, 2001) occasionnant de surcroît des délais supplémentaires aux projets logiciels. Même si une étude comparative récente des communications en contextes co-localisé et distribué (Al-Ani & Keith, 2008) lors de laquelle les chercheurs ont trouvé que les communications informelles se déroulaient aussi fréquemment en contexte co-localisé que distribué, ce qui amène ces derniers à se questionner s'il n'y aurait pas eu d'évolution en terme de pratique ou technologique depuis l'étude mentionnée ci-dessus, la méthodologie employée pour l'obtention des résultats dans le cadre de cette nouvelle étude et qui était basée sur les réponses à un questionnaire administré à 17 répondants et non le fruit d'observations empiriques sur le terrain porte à croire que de nouvelles études sur le sujet s'avèrent nécessaires.

1.4.3 Les solutions envisagées

Afin de pallier un tant soit peu aux divers problèmes susmentionnés, certaines pratiques peuvent être mises de l'avant. Entre autres, certaines technologies telles

que la vidéoconférence ainsi que la messagerie instantanée peuvent être mises en place afin de minimiser l'effet des distances géographiques (Herbsleb & Mockus, 2003-). De plus, la division et l'attribution des tâches d'un site à un autre devraient se faire tenant compte du concept de modularité du logiciel développé afin de minimiser les liens de dépendances entre les sites (Grinter et al., 1999). Ainsi, selon Grinter, Herbsleb et Perry, l'attribution des tâches pourrait se faire en fonction des domaines d'expertise de chaque site, de la structure du logiciel, des disciplines liées au processus de génie logiciel suivi, ainsi que du concept de configuration. Finalement, certaines méthodes liées à la gestion des connaissances et de l'expertise peuvent également être mises de l'avant, tel qu'un processus de gestion de connaissances institutionnalisées (Herbsleb & Moitra, 2001) de même qu'un fureteur d'expertises (Mockus & Herbsleb, 2002).

1.5 Les méthodes empiriques en génie logiciel

La recherche empirique, qui prend son fondement de la méthode expérimentale, est employée depuis des centaines d'années dans l'étude de divers domaines plus particulièrement liés aux sciences humaines telles que la psychologie, l'anthropologie ainsi que la sociologie. Elle est plus souvent qu'autrement considérée comme étant la seule méthode scientifique valable et acceptée dans ces domaines. En génie logiciel, plus jeune des disciplines du génie, la recherche empirique y est apparu depuis peu et à beaucoup plus petite échelle, mais tout de même, y est apparu depuis quelques années en raison, entre autres, de la popularité croissante des recherches ayant pour objet les divers aspects humains impliqués au cours de processus de génie logiciel (Seaman, 1999-; Hazzan & Hadar, 2008). Comme il a

été évoqué un peu plus tôt, de plus en plus de chercheurs soutiennent que nombre de problèmes rencontrés en génie logiciel ne sont pas tous imputables à des lacunes d'ordre technologiques, mais peuvent être également attribuables à certains facteurs humains intervenants au cours du processus de génie logiciel (Perry et al., 1994-; Seaman, 1999-). Ainsi, l'étude de l'Homme, loin d'être étudiée par la même approche que les divers aspects techniques auxquels sont habitués les chercheurs en génie informatique et génie logiciel, oblige ces derniers à élargir leurs méthodes de recherche et d'aller puiser certaines techniques et fondements théoriques de certains domaines liés aux sciences humaines.

Toutefois, bien que les méthodes empiriques commencent de plus en plus à faire leur marque dans le monde du génie logiciel, il y a encore beaucoup de réticences exprimées par plusieurs praticiens, et ce sentiment est encore beaucoup plus palpable à l'égard des sujets de recherche qui gravitent autour des aspects humains du génie logiciel. D'une part, parce que les aspects humains sont difficilement mesurables, et d'une autre part, parce que les méthodes pour les mesurer ne sont généralement pas bien maîtrisées des scientifiques plutôt habitués aux sciences pures. Cela est principalement dû au fait que l'étude des aspects humains requiert souvent une approche qualitative, ce qui va à l'encontre des sciences pures plutôt quantitatives, et vu l'état actuel de ces connaissances dans le domaine, la méthode inductive est souvent nécessaire au détriment de la méthode déductive afin d'être en mesure de générer les premiers fondements théoriques de ce domaine encore largement inexploré.

Cependant, comme le suggère l'avant propos, «tenter de mesurer le non-mesurable» est aussi important en génie logiciel que dans toutes autres disciplines et le «fait de proposer certaines mesures peut ouvrir un débat menant à une meilleure compréhension» du phénomène à l'étude (Fenton & Pfleeger, 1996). Par ailleurs, Tichy (Tichy, 1998) énonce, à ceux qui contesteraient encore cette nouvelle tendance, de nombreux arguments en sa faveur, par le biais de son article intitulé « Should Computer Scientists Experiment More? » De plus, de nombreux partisans de cette pratique en génie logiciel croient que maintenant que la quantité de recherches empiriques augmente, la qualité de ce type de recherche doit augmenter à son tour (Kitchenham et al., 2002; Walker, Briand, Notkin, Seaman, & Tichy, 2003). Conséquemment, le protocole de recherche suggéré et décrit dans le présent document fut largement influencé par les recommandations tirées des quelques publications susmentionnées.

1.6 Sommaire de la littérature

Lors de ce chapitre, différents éléments de littérature et aspects théoriques ont été explorés. Bien que le fil conducteur entre ceux-ci semble à première vue plus ou moins clair, il le deviendra sans aucun doute lors des chapitres suivants. Voici donc un bref sommaire de ce qui a été vu lors du présent chapitre et qui en donne un premier aperçu.

D'abord, il a été discuté de l'approche processus en génie logiciel et comment cette approche a permis à la science de connaître une importante évolution. L'importance de la mesure en génie logiciel a également été abordée et comment la mesure, tout

comme cette recherche, s'inscrivent dans une approche dite «bottom-up» permettant de faire de l'amélioration de processus.

Ensuite, des notions de base reliées au travail d'équipe en génie logiciel ont été rappelées pour après insister sur l'importance significative du travail collaboratif *ad hoc* réalisé en équipe dans le cadre d'un processus de génie logiciel, objet de cette recherche.

Afin d'être en mesure de bien comprendre les différents fondements psychologiques de la communication, et plus spécifiquement de la communication au sein des groupes, une revue sommaire des théories essentielles à garder en tête pour l'analyse des résultats de cette recherche a été réalisée.

Par la suite, le développement de logiciels dans un contexte distribué a été discuté. Les motivations d'affaires d'une telle pratique ont été énumérées, mais surtout, les problèmes de communications rencontrés ont été soulevés et les quelques pistes de solutions envisagées pour y palier ont été expliquées, rendant d'autant plus pertinent le sujet de la présente étude.

Finalement, la méthode empirique, approche utilisée dans le cadre de cette recherche a été introduite, méthode qui en est encore à ses premiers balbutiements dans le domaine du génie logiciel. La méthode imaginée pour mener à bien la présente étude fait d'ailleurs l'objet du chapitre suivant.

CHAPITRE 2

PROTOCOLE DE RECHERCHE

2.1 Énoncé du problème

Un processus de développement de logiciel classique se compose en un amalgame d'activités prescrites formellement et exécutées tantôt successivement, tantôt en parallèle, où l'extrant de l'un sert en partie ou en totalité à l'intrant de l'autre. L'essence de ce processus vise à cristalliser une quantité d'information de différentes natures en un logiciel qui devra répondre aux besoins de l'utilisateur dans les temps et les coûts prescrits, tout en respectant certains critères de qualité (Robillard et al., 2003). Le développement de logiciel est donc le fruit d'un travail mental et cognitif intense de la part de chaque individu participant à son développement afin d'emmagasiner, d'analyser et de comprendre la masse d'information nécessaire pour ensuite être capable de la synthétiser et de la transformer en différents artefacts requis à la réalisation du projet (Robillard, 1996). De plus, ces mêmes individus se retrouvant souvent regroupés en équipe afin d'unir les efforts essentiels à la réalisation du projet, une bonne collaboration entre ceux-ci s'avère donc indispensable afin de partager les informations nécessaires ainsi que de coordonner leurs efforts dans le but d'assurer le bon déroulement du projet.

Ainsi, le développement d'un logiciel dépend grandement du travail collaboratif réalisé par l'équipe de concepteurs en charge de mener le projet à terme. Il est

donc légitime de penser que la qualité de la collaboration entre les membres de l'équipe aura un impact inévitable sur la qualité du produit final.

Par ailleurs, tel qu'il a été discuté au cours de la revue de littérature, un certain nombre de chercheurs s'intéressent de plus en plus à l'aspect du travail collaboratif en génie logiciel. Plus particulièrement, certaines études réalisées dans ce domaine ont démontré qu'une large part du temps passé sur un projet de développement de logiciel l'est de façon *ad hoc*, par le biais de communication ou d'activités collaboratives informelles et spontanées (Perry et al., 1994-; Robillard & Robillard, 2000). Il semble donc très contradictoire de constater et de dénombrer une quantité d'effort relativement significative employée de façon *ad hoc* par opposition à la nature formelle des processus qui vise plutôt à prescrire des activités toutes autant formelles dans le but d'ordonnancer les efforts à fournir pour mener à terme le projet logiciel.

Il est, de ce fait, permis de se questionner sur les causes de cette non-compatibilité entre ce qui est «prescrit» formellement et ce qui est «observé» empiriquement. Peut-être y a-t-il justement une non-compatibilité entre les processus qui sont parfois inapplicables dans certains contextes, basés sur de nombreuses présomptions ou encore qui ne répondent pas à l'ensemble des besoins ou problèmes qui peuvent survenir lors du développement d'un logiciel, tel que l'ont déjà souligné certains auteurs (Briand et al., 1995; Bach, 1994), et les processus cognitifs sous-jacents des concepteurs de logiciels qui s'opèrent et entrent en jeu lors d'une activité précise.

Par ailleurs, comme il a été donné de constater dans le contexte de développement distribué de logiciels, la distance au sein des équipes virtuelles lève certaines barrières aux communications informelles résultant en divers problèmes de coordination et de partage de connaissances (Grinter et al., 1999; Herbsleb & Grinter, 1999; Herbsleb & Moitra, 2001). Cela démontre donc une fois de plus l'importance et la nécessité de ce type de collaboration.

Ainsi, les considérations susmentionnées, soutenues par les divers éléments vus au cours de la revue de littérature présentée dans ce document, mènent conséquemment à formuler la question de recherche suivante :

Qu'est ce qui caractérise le travail collaboratif *ad hoc* et qui explique qu'il constitue une part significativement importante du temps octroyé par les concepteurs de logiciels au cours d'un processus de génie logiciel, en dépit des nombreuses formes d'activités collaboratives formelles prescrites par de tels processus?

2.1.1 Objectifs généraux

Il est clair qu'une telle problématique de recherche ne trouve pas de réponse facilement et que la réponse à celle-ci n'est certainement pas non plus des plus simples. Toutefois, pour tenter d'opérationnaliser cette question en une démarche de recherche qui pourra peut-être y répondre et tirer potentiellement profit des résultats obtenus, les objectifs généraux et spécifiques de recherche suivants sont définis :

1. Mesurer et analyser le travail collaboratif *ad hoc* retrouvé au cours d'un processus de génie logiciel afin d'en identifier et caractériser les occurrences et le contenu;
2. Appliquer les résultats des observations réalisées en 1. pour ensuite formuler et proposer de nouvelles règles de l'art qui permettront de rendre les activités collaboratives réalisées entre les concepteurs de logiciel encore plus efficaces.

Ainsi, cette approche en deux grandes étapes adhère tout à fait à l'approche empirique prônant la compréhension d'un phénomène ou d'un problème avant de proposer quelconques interventions (Fenton & Pfleeger, 1996; Robillard et al., 2003).

2.1.1.1 Objectifs spécifiques

Afin de peaufiner un peu plus la démarche de recherche engagée, les objectifs généraux précédemment énoncés, et plus précisément en ce qui concerne le premier, peuvent se subdiviser en objectifs spécifiques suivants :

1. Observer le travail collaboratif dans le cadre d'une étude de cas en profondeur dans l'industrie pour essayer d'en dresser un modèle conceptuel théorique;

2. Caractériser les activités collaboratives *ad hoc* recensées, les communications qui en découlent et en identifier ainsi qu'en décrire le contenu;
3. Générer des hypothèses théoriques issues des résultats de cette recherche qui pourront être par la suite validées par diverses recherches confirmatoires.

Un premier volet à cette recherche se veut d'abord de nature exploratoire étant donné que le domaine à l'étude est toujours inexploré ou très peu s'il l'est déjà. En outre, les données empiriques recueillies offrent la possibilité d'induire une représentation théorique du travail collaboratif *ad hoc* retrouvé au cours du processus de génie logiciel observé. Par ailleurs, comme c'est le cas dans toutes études exploratoires, il est très important de garder un esprit d'ouverture quant à l'issue de cette recherche et à toutes découvertes potentielles pouvant y émerger, pour, d'une part, éviter de biaiser les observations et les résultats qui en découlent et d'une autre part, pour éviter d'omettre toutes pistes d'investigation éventuelles pouvant déboucher sur des résultats intéressants.

Le deuxième volet de cette recherche prend par la suite une tangente plus descriptive afin de pouvoir dresser, à l'aide du modèle conceptuel formé des données empiriques recueillies, un portrait plus précis des activités collaboratives *ad hoc* observées et des patrons comportementaux émergents.

Finalement, la suite logique à cette recherche se voudrait plutôt explicative en investiguant les raisons sous-jacentes aux résultats obtenus et ce, à l'aide de fondements théoriques issus de divers domaines, entre autres, des sciences humaines telles que la sociologie organisationnelle de même que la psychologie comportementale et cognitive. Des éléments de réponses en regard à ce dernier volet seront tentés dans le cadre de cette recherche, et ce, par le biais d'hypothèses induites des résultats obtenus, mais d'autres recherches explicatives devront toutefois être entreprises sur la base des inférences théoriques générées par cette étude afin d'en apporter un éclairage plus précis.

2.1.2 Pertinence théorique de la recherche

D'un côté théorique, il est permis de croire que de prime abord, l'importance qu'occupe le travail collaboratif *ad hoc* au cours des processus de génie logiciel, mais également l'absence ou sinon le peu de recherches empiriques réalisées et visant à décrire le contenu de ces activités collaboratives *ad hoc* justifient pleinement la pertinence de cette recherche. Celle-ci permet d'une part, d'établir un modèle de ce type d'activités, de mieux saisir les aspects cognitifs impliqués, à partir desquels il est possible dans un deuxième temps d'induire des hypothèses créant ainsi une base théorique dans ce domaine toujours inexploré du génie logiciel. Comme le suggère également une loi attribuée à Tom Gilb : «Anything that you need to quantify can be measured in some way that is superior to not measuring it at all...».

De plus, la pertinence théorique de cette recherche se justifie aussi dans le fait qu'elle permet d'approfondir, d'élaborer et d'appliquer des méthodes de recherches empiriques, exploratoires de surcroît, et ayant pour objet les facteurs humains impliqués au cours d'un processus de génie logiciel, domaine encore peu étudié en génie logiciel.

2.1.3 Pertinence pratique de la recherche

D'un côté pratique, une recherche telle que celle présentée dans le présent document est très pertinente pour plusieurs raisons. D'abord, les résultats découlant de cette étude peuvent servir à définir de nouvelles règles de l'art visant à apporter des améliorations au travail collaboratif en génie logiciel. Comme le propose Robillard :

« Il est possible qu'une connaissance des processus mentaux propres à une activité spécifique puisse aider à concevoir des règles de l'art qui soient appropriées et efficace pour cette activité. Il n'existe peut-être pas de démarche universelle applicable à toutes les situations, mais quelques approches adaptées seraient déjà un gain significatif. » (Robillard, 1996)

Partant du fait qu'une collaboration efficiente est une condition *sine qua non* pour que le travail d'une équipe de concepteurs de logiciels se fasse tout aussi efficacement afin d'obtenir un produit de qualité, qui correspond réellement aux besoins de l'utilisateur, dans les temps et les coûts prévus, cette recherche est pertinente d'un côté pratique puisqu'elle permet de proposer des améliorations aux processus de génie logiciel, améliorations qui seront d'autant plus adaptées à la réalité empirique du développement de logiciels. De plus, ces améliorations viseront également à répondre aux réactions divergentes et parfois contradictoires exprimés

dans la littérature, tel que soulevé au premier chapitre du présent document, face au travail collaboratif *ad hoc* et les communications informelles qui en découlent et permettront d'adresser ce type d'activités collaboratives de façon plus adaptée aux besoins inhérents qui les motivent.

Par ailleurs, comme il a été exposé au cours du chapitre portant sur le développement distribué de logiciels, les barrières aux communications informelles occasionnées par la distance au sein des équipes virtuelles provoquent entre autres, certains problèmes de coordination et de partage d'informations et de connaissances (Herbsleb & Grinter, 1999). Cela constitue donc une autre bonne raison pour laquelle il est très pertinent de comprendre le contenu de ces échanges informels dans une situation de développement standard où tous les concepteurs sont réunis sur un seul site et où ils ont toute la liberté de communiquer comme ils le veulent. Cette compréhension donne par la suite la possibilité de proposer, par exemple, certaines activités plus formelles de collaboration et offre également plus d'intrants quant aux besoins à définir en terme d'outils de communication à disposer en situation de développement distribué (Cherry & Robillard, 2004a).

2.2 Stratégie de recherche

La présente section vise à décrire la démarche de recherche suivie afin de répondre à chacun des objectifs de recherche susmentionnés.

2.2.1 Approche générale

Ce travail de recherche est réalisé au moyen d'une observation participante réalisée dans le cadre d'une étude de cas en profondeur en milieu industriel. L'observation participante est une méthode empirique très bien connue et largement utilisée en sciences humaines, particulièrement en anthropologie et en sociologie, dans le cadre de recherches qualitatives visant l'observation d'objets, d'évènements, de processus, de relations ou de personnes dans leur environnement naturel. Plusieurs raisons motivent le choix de cette méthode :

- l'observation des évènements et du phénomène à l'étude en temps réel;
- les données sont recueillies dans leur contexte (Seaman, 1999-);
- l'observation participante permet d'étudier les comportements et les relations interpersonnelles des participants dans leur environnement naturel;
- l'observation participante est appropriée pour les recherches exploratoires et descriptives (Babbie, 2001; Jorgensen, 1989);
- les échanges à observer ne sont pas visibles du point de vue d'un observateur hors du terrain d'observation, du public (Seaman, 1999-);
- le phénomène à observer n'est observable que dans la vie de tous les jours et difficilement reproductible dans un contexte expérimental;

- le phénomène est assez limité en étendu pour être observé sous forme d'étude de cas.

D'autre part, ce type d'approche est tout à fait approprié dans le cas de cette recherche puisque, tel que mentionné précédemment, cette dernière est exploratoire. De plus, comme le souligne Jorgensen (Jorgensen, 1989) et Babbie (Babbie, 2001), l'observation participante sur le terrain est appropriée lorsqu'il ne s'agit pas de recherche déductive ou d'expériences où le but est de vérifier empiriquement des hypothèses formulées à l'avance, mais plutôt dans le cas d'études visant à générer de façon inductive des théories à partir d'observations et de données empiriques collectées sur le terrain. De cette façon, les observations initiales peuvent mener à certaines conclusions, lesquelles pouvant guider certaines observations subséquentes, vérifiant les conclusions précédemment établies, impliquant peut-être une révision de ces dernières et ainsi de suite.

Bien que dans le cadre d'une observation participante en recherche exploratoire la définition du problème peut être le fruit d'un processus ouvert et flexible durant lequel ce qui sera étudié sera identifié, clarifié, raffiné et élaboré, même au cours des observations, donc après le commencement de l'étude (Jorgensen, 1989), la problématique et les objectifs de la présente recherche ont été établis à l'avance à partir des observations des études antérieures qui ont été exposées au cours du chapitre de revue de littérature. Toutefois, comme il a été mentionné précédemment, il est très important de garder un esprit d'ouverture quant aux

issues possibles de la recherche afin de ne pas biaiser ou orienter indûment les observations et les conclusions qui en découlent.

Par ailleurs, le rôle que choisi de revêtir l'observateur lors d'une observation participante est très important. Ce rôle pouvant varier dans un continuum d'observateur à participant complet, quatre types de rôle peuvent être définis :

- observateur complet où le chercheur ne participe aucunement aux activités du groupe observé;
- observateur-participant où le chercheur occupe davantage un rôle d'observateur que de participant, s'impliquant dans quelques activités du groupe observé;
- participant-observateur où le chercheur occupe davantage un rôle de participant que d'observateur, s'immisçant beaucoup plus dans les activités du groupe observé;
- participant complet où le chercheur participe pleinement aux activités du groupe observé, voire à en devenir un membre à part entière.

De plus, pour chacun des rôles décrits ci-dessus, deux stratégies d'entrée peuvent être privilégiées :

- ouverte, où le chercheur annonce clairement le but de sa présence aux observés;
- cachée, où le chercheur choisi de ne pas dévoiler ouvertement le but de sa présence. Toutefois, pour adopter cette stratégie d'entrée, le chercheur se doit d'avoir d'excellentes raisons d'agir ainsi puisque cette approche peut venir à l'encontre de certaines règles d'éthique fondamentales. Elle pourrait peut-être être valable, par exemple, dans le cas où le phénomène à observer risque de ne pas survenir si les sujets sont au courant du fait qu'ils sont observés. Cependant, le fait de participer à l'étude sans avoir préalablement donné son consentement ne doit pas comporter de risque ou d'inconvénient pour le sujet et ce dernier doit être mis au courant de sa participation à la recherche aussitôt qu'il est possible de le faire de même qu'il doit se voir également offrir le choix de refuser l'utilisation des données collectées le concernant pour analyse ultérieure.

Dans le cadre de la présente recherche, le rôle occupé par le chercheur est celui de participant complet et la stratégie d'entrée est ouverte. De plus, quoique les rôles de d'observateur et de participant soient généralement considérés en conflit étant donné qu'il est souvent véhiculé que plus la participation augmente, moins le chercheur est en mesure d'observer et vice-versa, cette dualité perd maintenant de son importance en observation participante puisqu'il est démontré que pour éviter les incompréhensions et les observations imprécises, il est impératif que l'observateur participant se rapproche aussi bien physiquement que socialement des observés (Jorgensen, 1989). Cela est également le cas dans la présente recherche

étant donné que pour comprendre le contenu des communications *ad hoc* qui se déroulent entre les concepteurs observés, le chercheur se doit de comprendre le vocabulaire et le jargon de ces derniers, la culture organisationnelle dans laquelle ils baignent, etc.

Un avantage intéressant de l'usage de l'observation participante dans le cadre de recherches exploratoires et d'études de cas telles que celle présentée dans le présent document est qu'elle permet de potentiellement générer des résultats hautement valides puisque la définition des concepts utilisés est puisée directement de la réalité empirique du monde des observés.

Toutefois, à cause de la nature subjective et ouverte de la méthodologie, le fait de devenir envahi par le phénomène implique le danger qu'il deviendra très difficile de faire des rétroactions afin de réévaluer les buts et la problématique de recherche de même que la direction à poursuivre. Ce risque est mitigé dans le cadre de la présente recherche de deux façons. D'abord, la part de subjectivité de la méthode est réduite au maximum à l'aide de techniques d'encodage qui seront détaillées plus loin dans ce document et cette part de subjectivité est ensuite estimée à l'aide de calculs de validité qui seront également exposés plus loin. Ensuite, le recul indispensable au chercheur afin de faire une rétroaction nécessaire à une bonne analyse sur les observations réalisées ainsi que leurs significations, sera soutenu par un second chercheur ayant conservé la distance nécessaire du terrain d'observations.

De plus, la conception conventionnelle de la distinction entre objectivité et subjectivité est plus nuancée en observation participante. D'abord, comme il a déjà été exposé ci-haut, il est essentiel de pouvoir accéder à la subjectivité du monde dans lequel vivent les observés afin de pouvoir faire des interprétations justes et véridiques des observations. Finalement, il est souvent essentiel d'expérimenter le phénomène à l'étude, de «devenir le phénomène» (Jorgensen, 1989).

2.2.2 Population cible

La population cible visée est constituée de toutes les équipes de développement de logiciels d'au moins deux personnes et plus, dans toutes organisations produisant des logiciels pour usage interne ou pour fins commerciales.

2.2.3 Échantillon et contexte de l'étude

L'échantillon, soit l'équipe de développement de logiciels sélectionnée, oeuvre dans une grande entreprise* productrice de logiciels pour fins commerciales, entreprise établie depuis plusieurs décennies et dans laquelle il existe un processus de développement clairement défini et que l'on pourrait qualifier de mature pour lequel elle a d'ailleurs obtenu une certification ISO 9001:2000.

Cet échantillon a été choisi tout simplement sur la base d'une opportunité de recherche offerte par l'entreprise participante, opportunité très rarissime en milieu

* En vertu d'une entente de confidentialité conclue avec l'organisation participante, son identité ne peut être divulguée.

industriel pour réaliser ce genre de recherche contraignant ainsi à saisir l'occasion sans pouvoir faire de choix sur une base plus scientifique ou statistique. Cependant, constituée de douze membres réguliers avec une échelle très variable d'âges, de degrés de scolarisation, d'années d'expérience en développement de logiciels ainsi que d'ancienneté dans la compagnie même, il est permis de croire d'un jugement relevant du sens commun (Babbie, 2001), que l'équipe observée peut être tout à fait représentative de la moyenne des équipes de développement qu'il est possible de retrouver dans l'industrie et en dépit du fait que cet échantillonnage s'est imposé un tant soit peu de lui-même. Bien que la recherche se déroule au sein d'une grande entreprise comptant plusieurs milliers d'employés, il est également permis de dire que celle-ci comporte également certains attributs de petites ou moyennes organisations étant donné la subdivision de sa structure organisationnelle en équipes de petites tailles pouvant compter quelques concepteurs à quelques dizaines d'individus. En outre, puisque cette recherche se déroule sous forme d'étude de cas et qu'à partir d'une étude de cas il est par définition impossible d'en généraliser les résultats à l'ensemble de la population, le choix de cet échantillon n'implique aucun biais notable.

Par ailleurs, voici quelques éléments d'information permettant d'avoir une image un peu plus précise du contexte dans lequel se déroule l'étude au sein de l'entreprise visée. D'abord, sur le plan organisationnel, tel qu'évoqués plus haut, huit concepteurs, dont un gestionnaire de projet, seront étudiés et ceux-ci s'intègrent au sein d'une équipe comprenant en tout 13 individus, soit un stagiaire travaillant avec les huit premiers et ayant certaines petites tâches spécifiques à mener à terme, et quatre autres concepteurs travaillant sur un autre projet. Par «concepteur» dans ce

contexte, il est entendu que ceux-ci sont responsables autant des phases d'analyse, de rédaction des spécifications, de conception du design que de l'implémentation du produit à développer. Toutefois, l'assurance qualité est assumée par une autre équipe ne faisant pas l'objet de la présente étude. De plus, cette équipe s'insère parmi plusieurs équipes de concepteurs de logiciels, ne travaillant pas nécessairement sur le même produit logiciel, au sein d'une instance de l'organisation basée à Montréal et qui comptent de nombreux autres centres de développement, notamment aux États-Unis, en Europe et en Asie. Concernant le produit, afin de bien comprendre le contexte de l'étude, il est pertinent de souligner l'énormité du logiciel développé comprenant des millions de lignes de code. Aussi, ce logiciel est constitué de plusieurs composantes, dont le développement est séparé en équipes dans lesquelles les membres d'une même équipe sont tous situés sur le même site, équipes distribuées cependant sur plusieurs continents. Étant donné l'interdépendance des composantes entre-elles en raison de la très grande complexité du domaine, ceci n'épargne pas les équipes à devoir interagir entre elles avec toutes les difficultés que la distance et les différents fuseaux horaires entraînent. Par ailleurs, non seulement le logiciel en lui-même est énorme, mais la complexité du domaine applicatif, telle qu'évoqué précédemment, ajoute une autre part de difficulté. Le concepteur se doit de comprendre une logique d'affaire parfois relativement complexe pour être en mesure de livrer un produit conforme aux spécifications des requis logiciels. Ensuite, la complexité des technologies impliquées ajoute également une autre part de difficulté, entre autres, étant donné l'architecture multicouche adoptée et n-tiers pour concevoir l'application de même que de l'emploi de plusieurs langages et paradigmes de programmation, sans compter des multiples gabarits et standards de programmation imposés.

2.2.4 Méthodes de collecte des données

La méthode de collecte de données suivante fut élaborée à la suite d'une période ethnographique de neuf mois au sein du groupe choisi. Non seulement celle-ci fut nécessaire afin de déterminer la méthode de collecte de données appropriée au terrain observé de même que les sources d'information à privilégier, mais elle permit également au chercheur de s'imprégner de la culture organisationnelle ainsi que d'acquérir le vocabulaire essentiel à l'analyse subséquente des faits observés. Aussi, tel qu'il a été mentionné ci-dessus, le rôle occupé par le chercheur lors de la phase de collecte de données était celui de participant complet et cela fut également le cas lors de la période ethnographique préalable. Cette expérience, augmentée également d'un certain nombre d'années préalables de pratique en entreprise sous divers rôles, concepteurs, architecte, gestionnaire de projets et d'équipe, donne au chercheur la maturité indispensable à l'analyse des données et à une interprétation de ces dernières qui va bien au-delà des faits observés.

Suite à cette période ethnographique, la méthode de collecte privilégiée fut de collecter le plus d'informations possibles de façon non-ciblée, même s'il faudra par la suite effectuer un échantillonnage sur la vaste quantité d'information amassée. Le but de cette approche est qu'étant donné que la recherche est de nature exploratoire et par conséquent, étant donné que les tangentes que celle-ci prendra en cours de route demeurent encore inconnues, il était préférable de collecter le maximum d'informations relatives au travail collaboratif *ad hoc* sur le terrain étudié

au cas où en cours d'analyse un élément d'information supplémentaire s'avèrerait nécessaire.

Ainsi, la phase de collecte de données s'est déroulée durant 37 jours ouvrables consécutifs. Ces 37 jours correspondaient pratiquement au temps complet d'un cycle d'évolution du logiciel développé par l'équipe observée. La collecte des données a donc été réalisée en cycle de maintenance, plutôt qu'en cycle de développement d'un nouveau logiciel et cet élément important doit être retenu et pris en ligne de compte lors de la phase d'analyse des données. Aussi, le but de la couverture complète du cycle d'évolution fait en sorte qu'il sera possible lors de la phase d'analyse, de pouvoir mesurer certaines variations aussi bien en terme de fréquence qu'en terme de contenu tout au long du cycle d'évolution observé.

Ainsi, les données recueillies lors de ces 37 jours de collecte de données incluent :

- plus de 190 heures d'enregistrements audio-vidéos de sessions de travail réalisées par les membres de l'équipe ;
- une copie des 2496 courriels reçus et envoyés par les huit membres de l'équipe observée ;
- une sauvegarde journalière du code source du logiciel développé ;
- une sauvegarde journalière de tous les documents et artéfacts internes se trouvant sur le serveur de fichiers de l'équipe.

Afin d'observer les échanges informels sous forme de conversations face à face entre les membres de l'équipe ainsi que les conversations téléphoniques de ces derniers, la méthode de collecte de données privilégiée fut l'enregistrement audio-vidéos de sessions de travail. Pour ce faire, une caméra vidéo et trois microphones étaient dispersés dans les espaces à bureaux, voir le schéma fourni en annexe, et 3 à 6 heures d'enregistrement étaient réalisées chaque jour et séparés en deux sessions, une l'avant-midi et l'autre l'après-midi. De plus, rien de particulier n'était demandé aux sujets, si ce n'est que de travailler comme ils le font habituellement et il n'y avait aucune intervention directe entre le chercheur (dans son rôle de chercheur étant également participant complet) et les sujets. Le but était justement de faire en sorte que les sujets vaquent à leurs occupations comme à l'accoutumé en oubliant la présence de la caméra et des microphones.

Ce choix a été préféré plutôt qu'à la prise de notes sur le terrain, tel qu'il est coutume de le faire en observation participante, et ce pour plusieurs raisons. D'abord, les enregistrements offrent la possibilité d'être visionnés à répétitions, ce qui constitue un énorme avantage dans ce cas-ci, étant donné que la recherche est exploratoire et que, par conséquent, ce qui est recherché n'est pas très bien défini d'avance. Il était donc tout à fait impossible de mener des observations plus ciblées. De plus, le fait qu'il soit possible de visionner à maintes reprises les enregistrements rend également possible, en cas de doute ou tout simplement pour aller chercher une information complémentaire, un retour en arrière autant de fois que désiré lors de l'analyse. Cette possibilité rend également réalisables certains tests de validité, comme il sera expliqué plus loin. Aussi, les enregistrements audio-vidéos

permettent de collecter une quantité beaucoup plus importante de données que s'il avait fallu tout prendre en note sur place au moment où les observations sont réalisées. Ainsi, avec les retours en arrière possibles et puisqu'il est également possible de prendre une pause lors de l'écoute des enregistrements, il est donc plus facile de faire ressortir une quantité beaucoup plus importante de données comparativement à des notes prises sur place. De plus, la caméra permettait l'enregistrement du travail de plusieurs sujets à la fois, alors qu'en situation de prise de notes sur le terrain, il aurait été impossible pour un observateur d'observer plusieurs sujets à la fois. Ainsi, l'apport de chaque heure d'enregistrement doit être multiplié par le nombre de sujets observés pour connaître le nombre d'heures équivalent qu'il aurait fallu faire si les observations avaient été réalisées directement sur le terrain. Finalement, contrairement aux notes prises sur le terrain, comme il a été le cas dans l'étude de Seaman et Basili (Seaman & Basili, 1997) ainsi que dans celle de Perry et Staudenmayer et Votta (Perry et al., 1994-) pour ne nommer que ces deux exemples, il apparaissait beaucoup moins intrusif de procéder par enregistrement audio-vidéos en plaçant une caméra discrètement dans un coin de la pièce plutôt que de rester assis directement aux côtés des participants en notant tous leurs faits et gestes. De la sorte, le risque était beaucoup moins élevé d'obtenir un biais dû au fait que les sujets se sentent épiés, tel qu'éprouvé entre autres par Ko, Deline et al. (Ko et al., 2007), et par le fait même d'obtenir des comportements beaucoup plus proche de la normale.

Toutefois, bien que les enregistrements audio-vidéos peuvent être considérés moins intrusifs, certaines précautions particulières ont été prises afin de minimiser les biais dus à la présence de la caméra. Premièrement, la caméra et les microphones ont

été placés de sorte qu'ils soient les moins visibles que possible, bien entendu dans la mesure où la qualité sonore des enregistrements ne s'en faisait pas trop ressentir. Aussi, des 37 jours d'enregistrements effectués, les 15 premiers jours de sessions enregistrées ne comptaient pas dans le cycle d'évolution observé et ont subséquemment été retranchées de l'analyse des données. Ce fait étant ignoré des sujets, cette période a donc permis aux observés de s'accoutumer de la présence de la caméra et des microphones diminuant ainsi les biais qui peuvent y être associés. D'ailleurs, certaines personnes ont même avoué qu'elles avaient oublié le fait qu'elles étaient filmées, ce qui ne se serait pas produit disaient-elles si un observateur avait été constamment à leur côté pour écouter leurs conversations.

La seconde méthode de collecte de données fut de faire la capture de tous les courriels transmis entre les membres de l'équipe ainsi qu'avec les intervenants externes avec lesquels ils travaillent. Cette source d'information semble très pertinente compte tenu du volume d'échanges relativement élevé. Ainsi, pour être certain de recueillir l'ensemble des courriels échangés, la sauvegarde des courriels se faisait de façon automatique et ce, à l'aide de règles configurées dans le logiciel de messagerie utilisé dans cette compagnie. Par ailleurs, cette capture incluait les courriels reçus et envoyés par chaque membre de l'équipe. Ainsi, cela permet d'effectuer une validation croisée des courriels recueillis, puisqu'un courriel envoyé par un émetteur doit nécessairement avoir été reçu par son récepteur et ce qu'un récepteur a reçu doit forcément avoir été envoyé par son émetteur. Toutefois, cette validation n'est possible seulement lorsque l'échange s'effectue entre deux membres de l'équipe puisque les courriels envoyés et reçus de la part des intervenants externes ne sont saisis qu'à partir des boîtes de messagerie des

membres de l'équipe. Cependant, le fait d'être capable d'appliquer cette validation aux membres de l'équipe permet d'estimer la validité de la saisie à l'ensemble de la collecte de courriels.

Finalement, la sauvegarde journalière du code source et des divers documents et artéfacts internes a été réalisée, d'une part puisqu'il était très facile de recueillir l'ensemble des artéfacts étant donné qu'ils se trouvaient tous sur un serveur de fichiers partagé et cette source d'information pouvait potentiellement servir à une analyse de contenu subséquente.

Plusieurs informations pour les besoins ont donc été récoltées lors de cette phase de collecte de données. Cependant, comme il a été mentionné auparavant, le but était d'amasser le maximum de données dans l'intervalle de collecte choisi et de faire un échantillonnage sur les données collectées plutôt que d'établir d'avance le type de données à ramasser, de ne recueillir que le strict minimum en courant le risque de se rendre compte lors de l'analyse qu'un élément d'information manque pour faire un type d'analyse. Cette dernière approche serait tout à fait pertinente dans le cas d'une recherche confirmatoire où la façon typique de procéder consiste à opérationnaliser les concepts que l'on désire mesurer, concepts qui sont connus à l'avance. Toutefois, dans le cas d'une recherche exploratoire telle que celle-ci, il était plus opportun de rester prudent et de garder toute la latitude de faire des choix appropriés au moment de l'analyse avec la flexibilité de changer de modèle de données ou de concepts si plusieurs itérations d'analyse s'avèreraient nécessaires. D'ailleurs, ces itérations sont très souvent un mal nécessaire dans le cas de

recherches exploratoires puisque ce qui est recherché n'est jamais très clairement connu d'avance.

Enfin, un aspect important des données collectées dans le cadre de la recherche présentée dans ce document, est leur appréciable qualité recueillies dans un contexte industriel bien réel, au sein d'une entreprise notoire et qui prêteront aux résultats obtenus toute la validité et la crédibilité souhaitée, dans la mesure où l'analyse de ceux-ci n'intègre pas de biais notable, et ce, plutôt que si la collecte avait été menée dans un contexte expérimental ou en milieu académique comme il est le cas dans nombre d'études retrouvées dans la littérature.

2.2.4.1 Considérations d'éthique

Comme dans toute recherche impliquant la participation de sujets humains, la question de l'éthique revêt une importance primordiale pour cette recherche. Toutefois, dans le cadre d'une observation participante, l'éthique de la recherche devient une préoccupation continue de tous les jours pour le chercheur. En effet, la conception de l'éthique est quelque peu différente, puisque les personnes avec lesquelles les observateurs participants interagissent ne sont pas considérées comme des sujets qui participent à une expérience ou une enquête. De plus, l'observateur participant a autant d'obligations d'éthique envers les individus sous étude que tout individu avec qui celui-ci interagirait dans la vie de tous les jours.

Ainsi, diverses mesures ont été mises en place afin d'assurer qu'aucun préjudice ne soit causé aux participants de l'étude. D'abord, on peut considérer que la balance

entre les inconvénients et les avantages rattachés à la participation de la recherche est équilibrée. En fait, aucun inconvénient direct n'est prévu de la participation à cette recherche et aucun avantage n'est également espéré, si ce n'est que les participants pourront prendre connaissance des résultats de la recherche qui pourra, il est permis de l'espérer, rendre le travail collaboratif plus efficace au sein de l'équipe.

De plus, une attention particulière quant à la collecte des données, de même que les traitements subséquents de ces dernières, a été portée pour qu'ils soient faits dans le plus grand respect à l'égard des sujets ayant participé à l'étude ainsi que des règles de l'art de rigueur dans le cadre d'une telle étude.

Par rapport à la capture des courriels transmis par les sujets dans le cadre de leur travail au sein de l'équipe, il est à noter que seuls ces derniers étaient sauvegardés et les courriels personnels échangés avec d'autres personnes, tels que des amis ou membres de la famille ne l'étaient pas. De plus, les sujets pouvaient à tout moment retirer les courriels qu'ils jugeaient confidentiels de leur dossier de courriels saisis, et ce, avant que l'observateur n'y ait accès.

Certains moyens ont également été mis en place afin de protéger l'anonymat des participants lors de la recherche ainsi que pour certifier la confidentialité des données. En effet, le nom des sujets fut, est, et sera remplacé par des codes d'identification dans toutes publications. De plus, le fichier contenant la liste des noms et leur code respectif, de même que toutes les données collectées sont sauvegardés en sécurité. Aussi, l'accès aux données est protégé puisque celles-ci

sont sauvegardées sur un disque dur externe ainsi que sur des disques optiques inaccessibles via réseau informatique ou depuis l'Internet et sont également rangées dans un casier verrouillé à clé à l'abri du vol.

Par ailleurs, afin que les sujets de l'étude puissent donner un consentement éclairé à leur participation à la recherche, une présentation orale de celle-ci fut donnée aux participants. Cette présentation exposait les buts poursuivis par cette étude, ce qui était attendu des sujets et des débouchés potentiels de la recherche. Les sujets avaient alors la liberté de poser toutes les questions qu'ils désiraient. Enfin, les participants ont tous signé un formulaire de consentement de participation éclairée dans lequel il leur était expliqué qu'à tous moments ils pouvaient se retirer de l'étude sans aucune question et sans aucun inconvénient ou préjudice que ce soit et que pour ce faire, ils n'avaient qu'à en informer un des deux chercheurs rattachés à l'étude. Les informations pour contacter ces chercheurs étaient d'ailleurs fournies dans le document, et après signature de celui-ci par les sujets, une copie de ce document leur a été remise.

Finalement, afin de certifier que cette étude se conforme à toutes les règles d'éthique établies, une demande d'évaluation du protocole de recherche a été dûment déposée au *Comité d'éthique de la recherche*, relevant du *Directeur de recherche et de l'innovation* et chargé de l'application de la *Politique sur l'éthique de la recherche avec des sujets humains* à l'École Polytechnique de Montréal. Après analyse du présent protocole de recherche par le comité, un certificat de conformité éthique, qu'il est possible de trouver en annexe du présent document, a été émis.

2.2.5 Méthodes d'analyse des données

La section qui suit décrira l'approche et certaines méthodes qui ont été mises de l'avant afin d'analyser la vaste quantité de données amassées, car cette étape constitue sans aucun doute la partie la plus fastidieuse et déterminante de la recherche. En effet, l'analyse consiste grossièrement à appliquer différentes techniques d'analyse sur les données empiriques recueillies sur le terrain afin d'obtenir un modèle conceptuel qui illustre le plus fidèlement possible la réalité empirique de ces données.

2.2.5.1 Analyse exploratoire et séquentielle des données

La principale approche d'analyse préconisée dans le cadre de la présente étude est issue de l'«analyse exploratoire et séquentielle des données», beaucoup plus connue sous le nom anglais de *Exploratory Sequential Data Analysis* (ESDA) (Fisher & Sanderson, 1993-). Cette technique d'analyse est toute indiquée pour les recherches exploratoires dont l'objectif est de trouver des réponses à des questions de recherches empiriques ou encore lorsqu'il s'agit de discerner des patrons parmi les données recueillies et de les décrire en utilisant, par exemple, de simples représentations statistiques. De plus, à partir de ces descriptions, la méthode de l'ESDA permet ensuite aux chercheurs d'énoncer certaines hypothèses qui seront par la suite validées par le biais de recherches confirmatoires utilisant des méthodes d'inférences statistiques. Cependant, la spécificité de l'ESDA réside dans le fait qu'elle s'applique plus particulièrement dans le cas de recherches où l'intégrité séquentielle des données doit être conservée. De ce fait, l'analyse de conversations,

d'interactions, l'analyse du verbal et du non verbal, des processus cognitifs ou de discours sont tous des exemples de domaines de recherches où l'information séquentielle des données est essentielle afin d'obtenir des résultats valables. Cette technique fut, entre autres, utilisée par Fisher et Sanderson (Sanderson & Fisher, 1994) dans le cadre de leurs recherches portant sur les interactions homme-machine, ainsi que par d'Astous et Robillard (Robillard et al., 1998; d'Astous, 1999; d'Astous & Robillard, 2000) dans leur étude visant à faire l'analyse des réunions de révision techniques au cours d'un processus de génie logiciel.

Tout d'abord, la technique de l'ESDA repose sur l'établissement et la définition de concepts formels issus de trois traditions ou influences (Sanderson & Fisher, 1994), concepts qui devront également prendre sens parmi les données empiriques observées. Ces influences, non seulement représentent en quelque sorte l'angle ou le point de vue avec lequel le chercheur analyse les données qu'il a récoltées, autrement dit, les aspects sur lesquels le chercheur portera attention au cours de son analyse, mais elles guident également l'approche à employer pour ce faire.

Ainsi, selon l'ESDA, la tradition behavioriste s'intéresse principalement à la description du comportement humain dans son environnement naturel. Les recherches qui sont menées suivant cette tradition tendent à être très objectives, quantitatives, recourant également à des méthodes d'échantillonnage statistique afin de faire le choix des individus à observer et pour finalement produire des résultats tout autant objectifs et répétables (Sanderson & Fisher, 1994).

La tradition cognitiviste s'intéresse essentiellement aux performances cognitives humaines dans l'accomplissement de certaines tâches. Les recherches sous cette influence se veulent plutôt explicatives. Cette tradition est d'ailleurs la suite logique aux recherches comportementales, puisque les recherches d'influence cognitiviste découlent souvent d'hypothèses formulées par les comportementalistes. Les études suivant la tradition cognitiviste descendent toutefois à un degré de granularité beaucoup plus fin qu'au niveau comportemental et s'intéressent à l'analyse du verbal. En fait, étant donné la difficulté à l'effet que les processus cognitifs ne puissent être observés directement, les cognitivistes assument souvent que les mots parlés représentent l'information située en mémoire à court terme à ce moment précis et peuvent donc servir à l'analyse des processus cognitifs sous-jacents qui s'y opèrent. Ces fondements sont par exemple ceux qui sous-tendent la technique bien connue du «think-aloud» (Sanderson & Fisher, 1994).

Enfin, la tradition sociale quant à elle se penche plutôt sur l'interaction et la communication de l'individu avec son milieu soit entre êtres humains ainsi qu'entre hommes et machines. D'ailleurs, les chercheurs suivant cette tradition procèdent souvent par observation participante en s'immisçant activement dans le milieu qu'ils étudient afin d'obtenir une meilleure compréhension de leur objet d'étude. Ils choisiront également d'étudier des situations représentatives et significatives du phénomène qu'ils étudient plutôt que de recourir systématiquement à des méthodes d'échantillonnage. De ce fait, cette tradition tend à être beaucoup plus qualitative que quantitative. De plus, la codification des données sous influence sociale est quelque peu différente dans le sens que l'encodage est souvent le fruit d'une interprétation plutôt que ce soit le produit de l'encodage qui serve à interprétation.

Ainsi, afin d'estomper les biais qui pourraient être associés à l'interprétation, les chercheurs suivant cette tradition recherchent souvent la participation des sujets sous étude au cours de leur analyse (Sanderson & Fisher, 1994).

Après avoir pris connaissance des principaux traits caractéristiques des traditions de l'ESDA tels que décrits par Fisher et Sanderson, il est essentiel de se positionner quant au point de vue qui sera adopté dans le cadre de ce projet de recherche. Tout d'abord, étant donné que les objectifs de recherche précédemment établis visent à caractériser le travail collaboratif *ad hoc* réalisé au cours d'un processus de génie logiciel, donc de prime abord, à dresser un modèle comportemental des concepteurs interagissant les uns avec les autres, et dans un deuxième temps, étant donné la nature exploratoire et descriptive de cette recherche, il est donc permis de déduire que celle-ci cadre très bien avec la tradition behavioriste. Cependant, tant l'objet à l'étude portant sur les interactions entre les concepteurs logiciels dans le cadre du travail collaboratif *ad hoc* étudié, que certaines approches employées pour en faire la mesure et l'analyse, comme celle d'adopter un rôle de participant complet dans le cadre de l'observation participante réalisée, font que cette recherche est fortement inspirée de l'influence sociale.

Maintenant, afin de comprendre un peu mieux le processus d'analyse de l'ESDA, ainsi que la démarche qui sera élaborée par après, il est d'abord essentiel de prendre connaissance des huit types d'opérations ou manipulations de données, baptisés «the eight Cs» par Fisher et Sanderson (Sanderson & Fisher, 1994). Ce sont ces manipulations qui serviront à faire l'exploration des données empiriques

recueillies et à en faire émerger un patron, une structure ou un modèle (Sanderson & Fisher, 1994) :

1. **segments;** consiste à grouper les données adjacentes qui démontrent ensemble une certaine cohérence. Ainsi, les données se retrouvent toutes regroupées en séquences qui sont ensuite contenues dans des séquences plus larges et ainsi de suite, afin de former un ordre hiérarchique dans les données;
2. **commentaires;** suggère simplement d'ajouter aux éléments de données des notes informelles ou formelles pour ajouter des précisions additionnelles ou certains éléments d'information pertinents supplémentaires aux données manipulées;
3. **codes;** sans doute la manipulation la plus importante, propose d'étiqueter chacun des éléments de données à l'aide d'un code formé d'une syntaxe particulière et contenu dans une liste exhaustive, exclusive et relativement restreinte de catégories et ce, afin de diminuer la variabilité des données de même que d'en faciliter la manipulation subséquente. C'est via cet encodage qu'il est possible de transformer les données qualitatives en données quantitatives rendant possible l'analyse statistique de celles-ci (Fisher & Sanderson, 1993-; Seaman, 1999-);
4. **interconnexions;** consiste à établir des relations entre des éléments de données qui ne sont pas nécessairement adjacents ou du même type;

5. **comparaisons;** comme son nom l'indique, il s'agit de la comparaison de différents éléments entre eux afin de comparer par exemple l'encodage des mêmes données brutes par différents analystes ou encore dans le but de comparer les données provenant de différents sujets, contextes ou conditions, entre un modèle de prédiction et les données empiriques réelles;
6. **limites;** est l'établissement de limites ou de bornes aux données afin d'en isoler une partie pour appliquer des transformations subséquentes;
7. **conversions;** suggère la transformation des données permettant ainsi l'émergence de nouveaux patrons;
8. **calculs;** propose de calculer des représentations de données sommaires telles qu'un simple décompte, une moyenne ou encore un test de signification statistique.

Ainsi, le processus d'analyse à l'aide de l'ESDA tel que proposé par Fisher et Sanderson (Sanderson & Fisher, 1994), est un processus itératif. Il consiste à définir une série de concepts issus des questions de recherche d'intérêt et basés sur les traditions de l'ESDA définissant le point de vue sous lequel le chercheur se placera pour analyser les données. Ces concepts guideront donc ce qui devra être observé parmi les données brutes collectées et quelles manipulations, telles que celles décrites précédemment, devront être effectuées sur ces données brutes afin d'obtenir des données dérivées sur lesquelles il est finalement possible de s'appuyer

afin de générer des théories et induire des hypothèses. Ce processus est itératif puisqu'il est souvent nécessaire de faire plusieurs retours en arrière par exemple, pour venir ajouter, enlever ou redéfinir des concepts ou catégories servant également à l'encodage des données.

Toutefois, bien que la description précédente présente une vue générale du processus d'analyse à l'aide de l'ESDA, ce processus peut être adapté à la tradition ESDA adoptée par le chercheur. De ce fait, la section suivante présentera la démarche spécifique d'analyse suivie dans le cadre du présent travail de recherche.

2.2.5.2 Démarche d'analyse suivie

Suite à la collecte de données décrite précédemment, une démarche d'analyse comprenant plusieurs phases a été entreprise afin de répondre à la question de recherche. D'abord, un échantillonnage des données à encoder a été réalisé. En premier lieu, une tranche des données récoltées comprenant les 15 premiers jours d'enregistrement a été supprimée au tout début de la phase de collecte de données. Cette suppression a été réalisée pour deux raisons. La première, parce que la phase de maintenance observée n'était pas débutée lors de ces jours. La seconde est que les sujets n'étaient pas au courant que les données récoltées lors de ces 15 premiers jours allaient être éliminées, alors ces jours ont donc permis aux sujets de s'accoutumer à la présence de la caméra. Cela permet ainsi de réduire les biais associés à la présence de la caméra et augmenter la chance d'observer des comportements plus naturels de la part des observés au cours des jours d'enregistrement servant à l'analyse. De ce fait, des 190 heures d'enregistrement

réalisées, 77 heures ont été ainsi supprimées laissant les 113 heures restantes disponibles à l'analyse. Ensuite, afin de réaliser les analyses sur des données ayant le plus de chance de représenter le travail collaboratif normal réalisé au sein de l'équipe, les jours d'enregistrement où au moins un sujet était absent ont été enlevés des heures d'enregistrement à analyser. Cette manipulation est justifiable puisque étant donné que l'objet de cette recherche porte sur le travail collaboratif entre les concepteurs de logiciels, il est permis de s'attendre à ce que le contenu de ce travail soit moins représentatif les jours où des coéquipiers sont absents. Ainsi, approximativement 44 heures ont été retranchées des 113 heures disponibles laissant un peu plus de 68 heures à l'analyse. Toutefois, après l'analyse des 27 premières heures d'enregistrement, il fut évident que le temps d'analyse était beaucoup trop long, alors la décision de réaliser un échantillonnage aléatoire systématique sur le reste des données fut prise. Ainsi, à partir du dernier enregistrement analysé à ce jour, un enregistrement était codifié à tous les trois enregistrements permettant ainsi d'éliminer 27 heures d'enregistrement et en laissant environ 41 heures à analyser. Étant donné que deux enregistrements étaient disponibles pour chaque jour d'observation, soit un le matin et l'autre l'après-midi, et étant donné le soupçon que la matinée pouvait être plus riche en travail collaboratif *ad hoc* que les après-midi, le soin a été pris d'alterner entre le matin et l'après-midi pour réaliser l'échantillonnage systématique. Par ailleurs, puisque le nombre d'enregistrements analysés est impair et que tous les enregistrements ne sont pas tous de la même durée, le nombre d'heures d'enregistrements analysés le matin et l'après-midi est inégal et il est important que cela soit pris en compte si des comparaisons entre ces deux groupes d'enregistrements sont effectuées lors de l'analyse.

Toutefois, des heures d'enregistrement analysées, il apparaît pertinent à ce point-ci de mentionner que trois sessions d'un total de 6 heures ont par la suite été également retranchées étant donné qu'elles étaient jugées non-représentatives. Dans le premier cas, une session appelée au sens où l'entendent Robillard et Robillard (Robillard & Robillard, 2000) s'est déroulée ce qui risquait de modifier complètement les interactions suite à cette réunion par rapport à la normale et dans le second cas, une très longue session de deux heures impliquant un des sujets observés a été dénotée ce qui, une fois de plus, risquait de biaiser les comportements des autres sujets étant donné que ce dernier était monopolisé toute la durée de l'enregistrement à une autre interaction et n'était donc pas disponible pour collaborer avec les autres. Dans le dernier cas, la session analysée a été retranchée étant donné le départ hâtif de certains sujets. Ainsi, des 41 heures d'enregistrement encodé, un total d'environ 35 heures d'enregistrement a donc été finalement conservé pour l'analyse. Les échantillonnages discutés ci-dessus sont illustrés sous forme d'un tableau présenté en annexe du présent document.

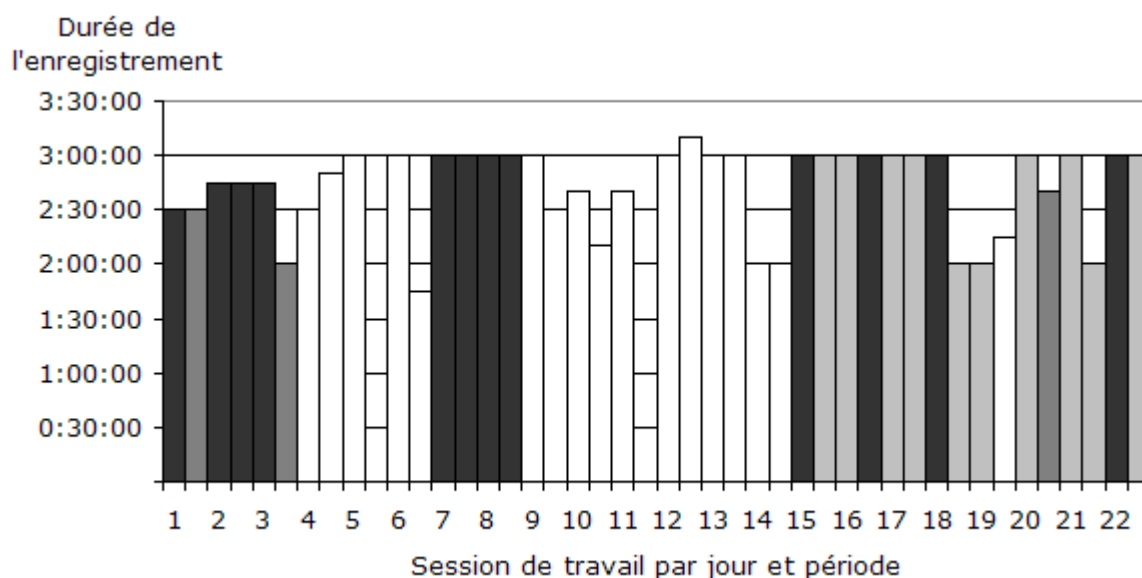


Figure 2.1 Durée d'enregistrement et échantillonnage des sessions de travail

La figure 2.1 ci-dessus illustre les enregistrements réalisés pendant les 22 jours d'observation. La première colonne de chacun des jours indique un enregistrement réalisé l'avant-midi alors que la seconde indique un enregistrement l'après-midi. Ainsi, en gris foncé sont illustrés les 12 enregistrements utilisés dans le cadre de l'analyse des données, alors qu'en gris moyen ceux ayant été éliminés une fois analysés, mais ayant été jugés non représentatifs. Les autres sessions disponibles à l'analyse et éliminées par l'échantillonnage sont représentées en gris pâle alors qu'en blanc figurent celles éliminées au départ en raison de l'absence d'au moins un des sujets.

Suite à ces échantillonnages réalisés sur les enregistrements à analyser, un second type d'échantillonnage a été appliqué sur les sujets pour lesquels le travail collaboratif sera analysé. Cependant, cet échantillonnage non aléatoire au jugé a

été basé sur des choix subjectifs. D'abord, sur les 12 sujets observés, quatre ont été éliminés de la liste des observés puisque même s'ils faisaient partie intégrante de l'équipe observée, ceux-ci travaillaient sur le développement d'un autre logiciel faisant en sorte qu'ils avaient beaucoup moins d'interactions avec les autres membres de l'équipe. Ainsi, des huit membres de l'équipe restant, pour une question de temps d'analyse, quatre autres membres de l'équipe ont également été retirés. Les quatre membres restants ont été sélectionnés puisqu'ils ont été identifiés lors de la période ethnographique ainsi que lors de la phase de collecte de données comme étant les quatre équipiers s'adonnant plus que les autres à du travail collaboratif avec leurs coéquipiers. Ce choix est justifiable étant donné que l'objectif de cette recherche n'est pas d'établir le temps passé par les concepteurs de logiciels au travail collaboratif *ad hoc* au cours d'un projet logiciel, l'importance de ce type d'activités a déjà été établie par diverses recherches relatées lors du chapitre de revue de littérature, mais le but est plutôt de chercher à comprendre le contenu de ce type d'activités. Or, il était donc permis de croire que de choisir d'observer les individus qui étaient les plus susceptibles de donner plus de matière à étudier était tout à fait défendable. De plus, comme il a été mentionné plus tôt, les chercheurs se plaçant sous la tradition sociale de l'ESDA tentent plutôt de réaliser des observations représentatives de la réalité empirique plutôt que de se référer systématiquement à des méthodes d'échantillonnage pour faire le choix des échantillons de données à analyser. Comme le soulignent Fisher et Sanderson :

«Researchers try to cover representative situations rather than conforming strictly to the requirements of sampling theory. Thus, material tends to be sampled on an ongoing basis-themes, artifacts, and agents are followed as they become relevant [...].» (Sanderson & Fisher, 1994)

Ainsi, une fois les données dûment échantillonnées, le processus suivi pour faire l'encodage des enregistrements sélectionnés est très similaire au processus générique de l'ESDA présenté plus tôt. Une représentation formelle des données, comprenant également l'établissement d'une liste de catégories servant à catégoriser les activités collaboratives *ad hoc* observées, est d'abord élaborée et sera détaillée au chapitre suivant. Ensuite, les données échantillonnées sont encodées conformément au formalisme d'encodage défini et les données qualitatives sous forme d'enregistrement où de courriels sont transformées en données quantitatives sur lesquels différentes analyses statistiques seront appliquées pour induire un modèle théorique du contenu des activités collaboratives *ad hoc* retrouvées au cours d'un processus de génie logiciel. Cette pratique est également connue sous l'appellation anglaise de «grounded theory» qui signifie le fait de générer des théories prenant fondement parmi les données empiriques recueillies (Babbie, 2001; Seaman, 1999-). La figure suivante, où les flèches à double-sens représentent les itérations ou retours en arrière souvent nécessaires, illustre le processus général d'analyse de données à l'aide de l'ESDA.

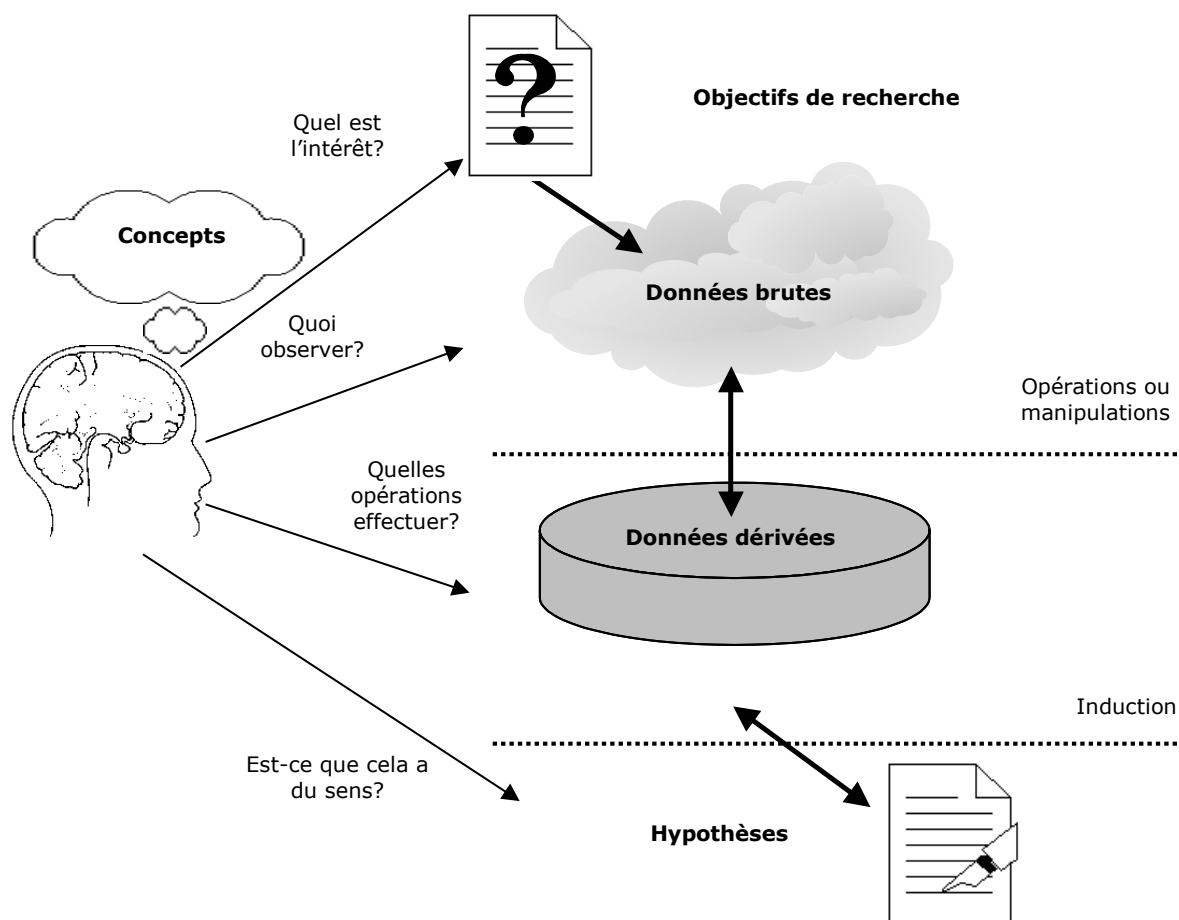


Figure 2.2 Processus d'analyse à l'aide de l'ESDA
Inspiré de Fisher et Sanderson (Fisher & Sanderson, 1996)

L'un des aspects primordiaux à tenir en compte dans le cadre de recherches exploratoires à l'aide du processus d'analyse de l'ESDA est l'aspect temporel de la recherche soit le ratio TA :TS où TA désigne le temps d'analyse requis pour un temps de séquence TS donné. Ainsi, selon Fisher et Sanderson (Sanderson & Fisher, 1994), le ratio TA :TS varie normalement de 5 :1 à 100 :1, mais des ratios plus extrêmes sont également possibles. L'un des facteurs qui fera diminuer ou augmenter considérablement le ratio TA :TS est le niveau de granularité recherché

par le chercheur lors de l'analyse des séquences de données brutes. Comme le rapportent les auteurs, il est possible de trouver dans la littérature des ratios aussi bas que 3 :1 dans le cadre de recherches portant sur l'analyse de l'utilisabilité de logiciels à partir d'enregistrement vidéos, alors qu'en contraste, il est également possible de trouver des ratios allant jusqu'à 500 :1, voire 5000 :1 dans le cadre de recherches portant sur les processus cognitifs d'utilisateurs de logiciels.

Dans le cas du présent projet de recherche, l'adoption de certaines mesures contribue à faire diminuer le ratio TA :TS. D'abord, la période ethnographique préalable a facilité à établir la représentation formelle des données de même qu'une première définition des concepts utilisés pour l'encodage des données. Ainsi, moins d'itérations se sont avérées nécessaires pour peaufiner le formalisme utilisé. De plus, la conception d'un outil spécialement conçu pour les besoins d'analyse de ce projet de recherche à l'aide de l'ESDA et permettant la saisie de données de même que l'analyse automatisée de celles-ci contribue certainement à accélérer certaines tâches qui autrement auraient été exécutées manuellement. Des images écran de cet outil sont présentées en annexe du présent document.

2.2.5.3 Représentation formelle des données

La représentation formelle des données est l'une des étapes les plus importantes de ce projet de recherche. En fait, non seulement la définition même du formalisme constitue un défi de taille étant donné que par définition, au cours d'une recherche exploratoire, le phénomène observé n'est pas encore très bien cerné au début de

l'étude, mais l'encodage des données à l'aide de ce formalisme fera en sorte que tous les résultats et conclusions qui en découleront en seront fortement influencés.

Toutefois, avant de définir la représentation formelle des données, il est important de définir l'unité d'analyse qui sera utilisée pour analyser les données recueillies. Ainsi, étant donné que l'objet de ce projet de recherche porte sur le travail collaboratif *ad hoc* réalisé entre les concepteurs logiciels et que ce type de travail collaboratif est principalement observable au cours des enregistrements vidéos recueillis sous forme de conversations verbales ayant lieu entre les intervenants qui prennent part à ces activités collaboratives, il y a donc lieu de se référer à la théorie de la linguistique pour aider à définir l'unité d'analyse appropriée dans le cadre de cette étude. Pour ce faire, la définition des unités observables lors d'une conversation proposée par Kerbrat-Orecchioni (Kerbrat-Orecchioni, 1998) constitue une excellente structure de repère. Kerbrat-Orecchioni définit les cinq unités conversationnelles suivantes :

1. acte de langage : plus petite unité monologique constituée d'un contenu et d'une forme illocutoire;
2. intervention : unité monologique composée d'un ou plusieurs actes de langage de la part d'un locuteur contribuant à un échange particulier;
3. échange : plus petite unité dialogale composée d'au moins deux interventions de locuteurs différents;

4. séquence : bloc d'un ou plusieurs échanges reliés par un fort degré de cohérence sémantique ou pragmatique, c'est-à-dire traitant d'un même thème ou centré sur une même tâche;
5. interaction : unité communicative d'une ou plusieurs séquences présentant une évidente continuité interne (participants, cadre spatio-temporel) et rompant avec ce qui la précède et la suit.

Ainsi, les interactions constituent l'unité communicative de base à une analyse conversationnelle. Elles sont constituées de séquences qui permettent de borner par thèmes les échanges d'une même interaction. Enfin, les interventions et les actes de langage permettent de distinguer l'apport d'un locuteur à un échange.

Bien que cette structure permette de réaliser une analyse verbale très microscopique des interactions observées, il serait tout à fait inutile, dans un premier temps, de s'aventurer sur une telle piste aveuglément, ne disposant même pas d'un portrait macroscopique du travail collaboratif *ad hoc* réalisé par les concepteurs au cours d'un processus de développement de logiciel. De ce fait, les unités monologiques, définies plus haut, de même que le niveau des échanges, sont des unités beaucoup trop détaillées pour servir à l'analyse des données de ce projet de recherche. Finalement, l'analyse des interactions composées de leurs séquences semblent être toute indiquée afin de dresser un portrait comportemental du travail collaboratif réalisé par les concepteurs de logiciels dans le cas étudié et permettront certainement d'en décrire un peu mieux la teneur. Toutefois, étant donné que certaines interactions observées peuvent aussi bien prendre une forme monologique

que dialogale, la représentation formelle des données qui suit devra également être assez flexible pour tenir compte de cette réalité. La figure suivante illustre donc le modèle de données retenu.

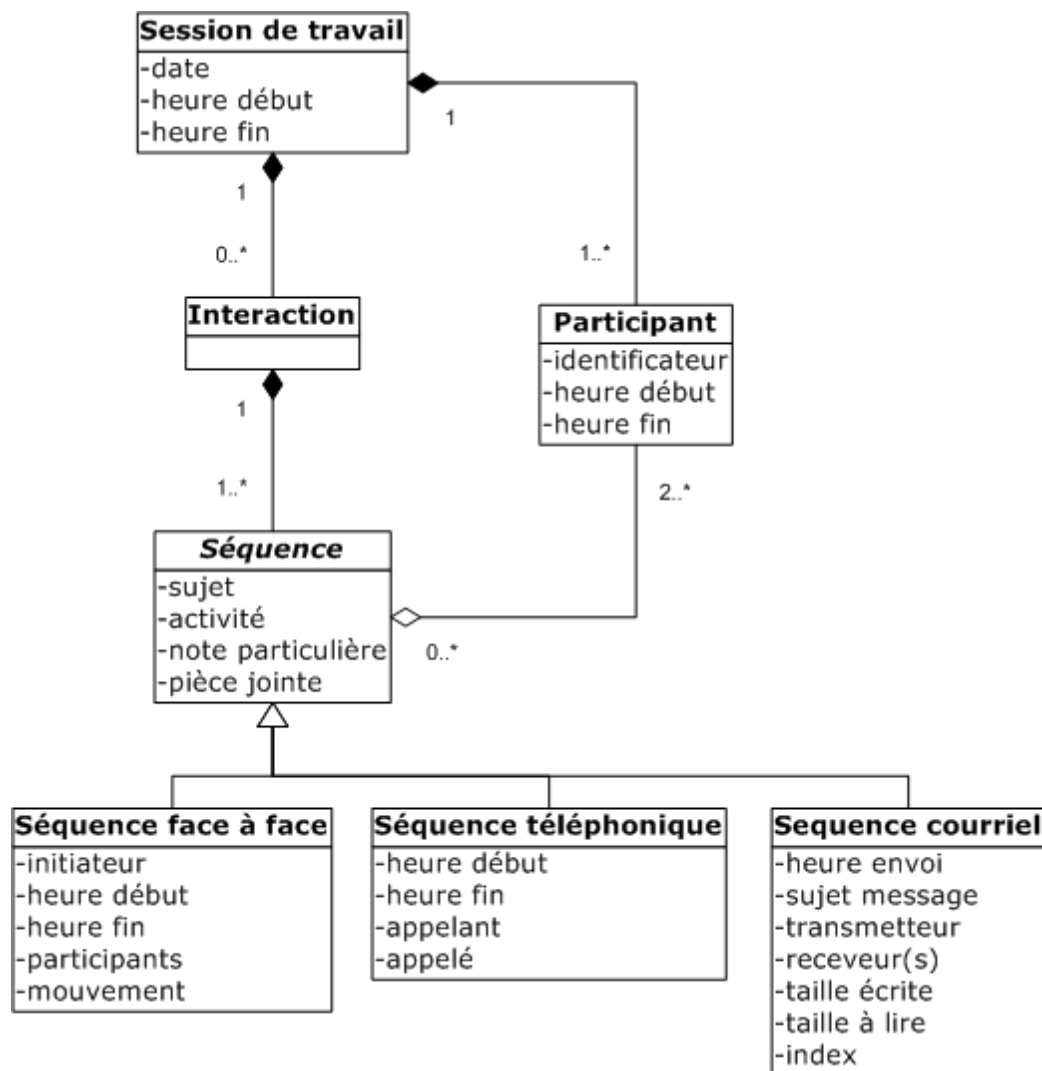


Figure 2.3 Diagramme orienté-objet du modèle de données

Le diagramme ci-dessus, représenté à l'aide du formalisme UML, est donc un amalgame de linguistique et du paradigme orienté-objet, tel qu'il fut inspiré par d'Astous et Robillard (Robillard et al., 1998; d'Astous & Robillard, 2000; d'Astous,

1999) lors de leur étude des réunions de révision technique du processus de génie logiciel. De plus, étant donné que le paradigme orienté-objet et le formalisme UML sont deux notions maintenant très largement répandues dans le domaine du génie logiciel, ces deux concepts ont été retenus afin de représenter le modèle de données utilisé dans le cadre du présent projet de recherche, mais ils ne seront toutefois pas développées dans le présent document. Cependant, les notions nécessaires à la bonne compréhension du diagramme peuvent être acquises par le biais de tous bons livres de modélisation en orienté-objet tels que le livre intitulé *Applying UML And Patterns* de Craig Larman (Larman, 2002).

Ainsi, comme le suggère la figure, une première classe d'objets a été définie afin de représenter les «sessions de travail» qui seront observées parmi les enregistrements vidéos recueillis. Cette information doit donc inclure la date de même que l'heure du début et de la fin de la période analysée. Étant donné qu'il est possible qu'un participant observé soit absent en partie de la période analysée, que ce soit pour un retard ou parce qu'il décide de quitter le travail plus tôt, il serait également pertinent de garder une trace de cette information afin d'en tenir compte au cours de l'analyse. La classe «Participant» incluant l'identificateur du participant ainsi que l'heure du début et la fin de sa présence représente donc cette information. Un lien de composition est également créé entre les deux avec une multiplicité de 1 à plusieurs puisque au moins un participant doit être présent pour qu'une session de travail soit analysée et l'instance de l'objet représentant cette participation est dépendante de la session de travail à laquelle elle se rattache.

Par ailleurs, composant les sessions de travail, la classe «Interaction» représente une interaction au sens où l'entend Kerbrat-Orecchioni (Kerbrat-Orecchioni, 1998). Cette entité représente, aux fins de l'encodage des données de ce projet, une interaction *ad hoc*, autrement dit, un travail collaboratif *ad hoc* qui a cours entre deux ou plusieurs participants. Toutefois, cette entité ne comporte pas d'attribut propre à elle, étant donné que les données la représentant pourront être dérivées des séquences la composant. Par exemple, pour déterminer l'heure du début de l'interaction, il ne suffira qu'à consulter l'heure du début de la première séquence qui la compose. Pour déterminer l'heure de la fin de l'interaction, il ne suffira également qu'à regarder l'heure de la fin de la dernière séquence. De même, pour identifier l'initiateur de l'interaction, cette information pourra être obtenue également à partir de la première séquence, etc... Ainsi, comme il vient d'être évoqué, les interactions sont composées de séquences, toujours au sens où l'entend Kerbrat-Orecchioni (Kerbrat-Orecchioni, 1998). Toutefois, contrairement au lien de composition entre la classe «Session de travail» et la classe «Interaction» qui indique qu'il est possible qu'aucune interaction ne survienne lors d'une session de travail, le lien de composition unissant la classe «Interaction» à la classe «Séquence» spécifie qu'au moins une séquence doit composer l'interaction pour qu'une instance de cette dernière existe. Aussi, un lien de partage uni la classe «Séquence» et la classe «Participant» spécifiant qu'au moins deux participants doivent prendre part à une séquence et que ceux-ci peuvent prendre part à aucune, une ou plusieurs séquences observées.

Enfin, la représentation des différents types de séquences pouvant survenir tout dépendamment du canal de communication utilisé est illustrée dans le diagramme à

l'aide d'une structure d'héritage. D'abord, la classe «Séquence» regroupe tous les attributs de données pertinents à tous les types de séquences. De plus, le choix de représenter cette classe sous forme de classe abstraite indique qu'une séquence se doit d'être d'un type particulier, c'est-à-dire, d'une instance de ses sous-classes. Ainsi, les attributs propres à tous types de séquences sont le sujet (thème) de la séquence, le type d'activité réalisé, une note particulière pouvant servir à l'analyse ainsi qu'un fichier joint qui peut être un artéfact sur lequel travaillent les sujets ou encore le contenu d'un courriel transmis, etc.

Pour ce qui est de la classe «Séquence face à face», cette dernière représente les activités collaboratives qui ont eu lieu en face à face verbalement et qui ont pu être observées lors de l'écoute des enregistrement vidéos. Les informations pertinentes propres à ce type de séquences sont l'identificateur du participant ayant initié la séquence, l'heure du début et de la fin de la séquence, la liste des identificateurs et des participants y ayant pris part, s'il y a eu mouvement lors de la séquence, c'est-à-dire si l'un ou l'autre des participants a eu à se déplacer de son espace de travail pour aller au bureau d'un autre, et si tel est le cas, quel est l'identificateur du participant au bureau duquel la séquence s'est déroulée. La classe «Séquence téléphonique» comme son nom l'indique représente une séquence ayant eu cours par l'entremise d'un entretien téléphonique. Les données s'y rattachant sont l'heure du début et de la fin de la séquence ainsi que l'identificateur de l'appelant et celui de l'appelé. Finalement, la classe «Séquence courriel» formalise une séquence s'étant déroulée par le biais d'un courriel et comporte l'heure de l'envoi, le sujet du message, l'identificateur du transmetteur, ceux des receveurs, la taille écrite par le transmetteur, la taille à lire par les receveurs, de même que l'index identifiant le

message au sein d'un fil de messages portant sur le même sujet. Il est important de mentionner ici qu'étant donné que l'effort de lecture et écriture de courriels est relativement difficile à discerner au visionnement des séquences vidéos, l'effort déployé à l'écriture et à la lecture d'un courriel sera établi sur la base du nombre de mots. La distinction entre le nombre de mots écrit ou à lire sert à éliminer le biais dû au copié collé de longs pans de texte, par exemple de traces d'exécution, etc.

Tous les attributs indiqués ci-dessus sont de type scalaire mis à part la liste des participants d'une séquence donnée. Cependant, la plupart des types de données sont énumératifs. Bien que les dates et heures des sessions de travail doivent être comprises dans la période lors de laquelle la collecte de données eut lieu, celles des interactions dans celles de sessions de travail, etc., les attributs représentant des identificateurs de participants ne doivent contenir que les participants ayant été identifiés lors des observations. Les sujets des séquences devront être compris également dans un ensemble de thèmes ouverts ayant été recensés lors des observations et finalement, l'attribut représentant le type d'activité menée lors d'une séquence doit contenir une des catégories d'activités collaboratives *ad hoc* identifiées et définies ci-dessous :

partage d'information : survient lorsqu'un participant partage une information avec un ou plusieurs autres sujets de façon asynchrone (i.e. sans que ce dernier en ait fait la demande), contrairement à la synchronisation cognitive. Par exemple : «En passant Michel, j'ai

modifié un bout de ton code dans la classe XYZ pour que ça fonctionne dans telle condition.»

synchronisation cognitive : survient lorsque deux ou plusieurs participants échangent de l'information afin d'être sûr qu'ils détiennent la même connaissance ou partagent la même représentation de l'objet dont il est question. Cette activité se distingue souvent par une suite de questions et réponses de part et d'autre. Par exemple : Sujet 1 : «Quelle est le but de la fonction ABC dans la classe DEF?» Sujet 2 : «Elle te retourne la configuration HIJ.».

développement : survient lorsque deux ou plusieurs participants s'adonnent au développement d'une nouvelle fonctionnalité, particularité ou composante du logiciel ou à la résolution d'un problème. Sujet 1 : «Si j'étais toi, au lieu d'implémenter ça comme tu l'as fait, j'utiliserais la librairie ABC pour le faire. De la sorte, tu appelles, la fonction XYZ et tu...»

coordination : survient lorsque deux ou plusieurs participants discutent afin de coordonner et planifier des activités telles que des réunions, des séances de travail

communes ou encore pour fixer des échéanciers. Par exemple : Sujet 1 : «Quand serais-tu disponible pour se rencontrer à propos de ça?» Sujet 2 : «Après dîner.» Sujet 1 : «Ça marche pour moi.»

non-pertinent : survient lorsqu'il n'est pas question du projet ou encore du logiciel à développer. Par exemple : Sujet 1 : «Es-tu allé voir le nouveau film XYZ en fin de semaine dernière au cinéma?»

Comme il a déjà été exposé précédemment, il est primordial que la définition des catégories d'activités, comme celle énumérée ci-dessus, se fasse à partir d'observations sur le terrain. Toute définition préalable aurait comme conséquence de donner une idée préconçue de ce qui peut être trouvé et fausse le point de vue de l'observateur pouvant le mener à une mauvaise compréhension du phénomène. Cela est également un aspect important de validité qui sera abordé plus tard. De ce fait, les définitions ci-dessus ont été élaborées à la suite de la période ethnographique qui a précédé la collecte de données.

Toutefois, bien que cette liste de catégories se veut d'être la plus exhaustive que possible et les catégories qui y figurent se veulent d'être aussi exclusives que possible, la catégorisation des activités observées demeure une étape importante de ce projet en raison de la difficulté à définir des règles de catégorisation très précises laissant plutôt le soin au codeur d'user de son bon sens pour trouver la bonne étiquette à son observation. Cependant, comme il fut expliqué lorsque la tradition

sociale de l'ESDA fut décrite, l'encodage sous l'influence sociale tend à refléter une certaine interprétation de la part du codeur plutôt que d'encoder des données qui seront par la suite interprétées. Comme le soulignent Sanderson et Fisher (Sanderson & Fisher, 1994) :

«Coding, when performed, is usually approached quite differently from other traditions. Codes tend to already reflect interpretation rather than be abstractions relying on processing for their meaning to emerge. Moreover, social researchers recognize that multiple valid interpretations of observational data exist. Because researchers try to infer the meaning of raw events in the face of these possible interpretations, codes sometimes appear more like final research statements than data points [...]. Although the goal may be to uncover the internal organization of phenomena, interpretational bias is acknowledged and explored, rather than eliminated, because it is held that no neutral point of view can exist. Thus, there is room for greater involvement of the participants themselves in the interpretation and analysis of the data. Altogether, bias is often managed quite differently from how it is managed in the other traditions.»

Ceci étant dit, bien que cette approche puisse mener à certaines interprétations qui pourraient diverger d'un codeur à un autre, certaines mesures visant à tester et accroître la validité des données peuvent être mises en place et seront exposées au cours du chapitre portant sur la validité et la fiabilité de la recherche.

Par ailleurs, afin de sérialiser les données analysées et les encoder sur support informatique physique, le formatage de celles-ci est réalisé à l'aide du format XML (Extensible Markup Language) (World Wide Web Consortium (W3C), 2005a). Plusieurs raisons motivent ce choix. D'abord, non seulement le format XML permet de représenter les données sous une forme intelligible pour l'être humain, il peut être également lu par n'importe quelle plateforme logicielle dans un très vaste éventail de langages informatiques. De plus, la lecture des fichiers en format XML

par le biais d'un analyseur syntaxique permet optionnellement de valider ceux-ci à l'aide d'un schéma XML de spécification de structure de document (World Wide Web Consortium (W3C), 2005b). Cette validation permet donc de vérifier si la structure du document est conforme à la spécification, spécification qui dans ce cas-ci reflète la structure de données spécifiée en format UML plus haut. Cela permet alors d'éliminer une large part des erreurs qui pourraient s'être glissées accidentellement lors de l'encodage des données. Le schéma XML utilisé dans le cadre de cette recherche est illustré en annexe du présent document.

Ainsi, de l'encodage des données à partir de la structure de données spécifiées ci-dessus et de leur sérialisation en format XML suivant les spécifications du schéma XML, on obtient un résultat s'apparentant à l'échantillon suivant.

```

<interaction>
  <sequence-list>
    <face2face-sequence initiator="MS1">
      <start-time year="2003" month="9" day="2"
        hour="9" minute="7" second="2"/>
      <end-time year="2003" month="9" day="2"
        hour="9" minute="7" second="25"/>
      <stakeholders list="MS1 MS2"/>
      <movement place="MS1"/>
      <purpose subject="Development Process"
        activity="Cognitive Synchronization"/>
      <particular-note>...</particular-note>
    </face2face-sequence>
    <face2face-sequence initiator="MS2">
      <start-time year="2003" month="9" day="2"
        hour="9" minute="7" second="26"/>
      <end-time year="2003" month="9" day="2"
        hour="9" minute="7" second="36"/>
      <stakeholders list="MS1 MS2"/>
      <movement place="MS1"/>
      <purpose subject="Configuration Management"
        activity="Information Sharing"/>
      <particular-note>...</particular-note>
    </face2face-sequence>
  </sequence-list>
</interaction>
<interaction>
  <sequence-list>
    <email-sequence initiator="MS2">
      <sent-time year="2003" month="9" day="2"
        hour="9" minute="11" second="0"/>
      <stakeholders sender="MS2"
        receivers="AI22 AI23"/>
      <purpose subject="Configuration Management"
        activity="Cognitive Synchronization"
        message-subject="..."/>
      <attachment path="..."/>
      <written-size nbwords="20"/>
      <to-read-size nbwords="32"/>
      <index value="5"/>
    </email-sequence>
  </sequence-list>
</interaction>

```

Échantillon 1 Exemple type de données utilisées pour l'analyse

2.2.6 Considérations de validité et fiabilité de la recherche

2.2.6.1 Validité

La validité des données réfère à la propriété selon laquelle la mesure empirique traduit fidèlement la réalité empirique du phénomène mesuré (Babbie, 2001). Ainsi, au cours de la stratégie de recherche proposée, certaines mesures sont mises de l'avant afin d'éliminer le plus de biais que possible et d'assurer une certaine validité des résultats.

D'abord, le premier aspect de validité à considérer est de s'assurer d'une certaine couverture des données concernant le phénomène d'intérêt ou autrement dit, les données récoltées devront être assez significatives et représentatives de l'objet à l'étude dans le cadre de ce projet. Cet aspect de validité a été adressé par le biais de la période ethnographique qui a précédé la phase de collecte de données et qui a permis d'identifier les différentes sources de données permettant de recueillir le maximum d'informations pertinentes quant au sujet d'étude.

De plus, afin d'accroître la validité de la recherche, les définitions des concepts ou des catégories utilisées dans le cadre de l'encodage ainsi que de l'analyse doivent prendre racine parmi les traditions de l'ESDA d'intérêt ainsi qu'à partir du contexte du terrain sur lequel s'effectue la recherche. Ce point est également adressé par le biais de la période ethnographique et du rôle du participant complet dans le cadre de l'étude, deux approches qui permettent d'accéder de très près à la réalité empirique des observés. Cette mesure permet donc d'aspirer à une plus grande

représentativité des résultats face au phénomène à l'étude par un aspect de validité référé dans la littérature comme relevant du sens commun (face validity) (Babbie, 2001; Jorgensen, 1989).

Aussi, les concepts ou catégories à partir desquels sont encodées les données de même que le nombre de catégories choisies sont également un point important en ce qui a trait à la validité de contenu (content validity) (Babbie, 2001). Cet aspect de la validité réfère plutôt, quant à lui, à la couverture des sens couverts par les concepts. Par exemple, catégoriser tous les échanges collaboratifs observés sous une seule ou deux catégories serait faire preuve de réductionnisme et cacherait tous les concepts sous-jacents servant à saisir la complexité du travail collaboratif *ad hoc* observé. Par ailleurs, la validité des interrelations (construct validity) (Babbie, 2001) doit être mesurée et assurée entre les concepts formant les patrons qui peuvent émerger des données empiriques recueillies. Cela peut être contrôlé par le biais de certaines mesures de corrélations ou d'associations statistiques, par exemple en formant une matrice de transitions ou de changements d'états entre les différents concepts pour démontrer la fréquence du passage d'un certain état ou d'un concept à un autre.

Finalement, afin de pouvoir accroître la plupart des aspects de validité interne susmentionnés (Seaman, 1999-), une triangulation de données doit être effectuée entre résultats quantitatifs obtenus et les données qualitatives observées de même qu'entre des données provenant de différentes sources (Walker et al., 2003).

2.2.6.1.1 Mesure d'accord intra et inter-codeurs

La mesure d'accord intra et inter-codeurs représente une étape déterminante du présent projet de recherche puisque cet important aspect de validité permet d'établir la solidité des résultats, c'est-à-dire que ces résultats peignent réellement la réalité qu'ils sont censés représenter. La mesure d'accord peut cependant être également considérée comme une mesure de fidélité ou fiabilité dans le sens qu'elle permet d'attester ou non que les résultats obtenus sont répétables.

Comme il a été mentionné un peu plus tôt, l'encodage des données constitue certainement l'opération la plus importante, et ce, pour deux principales raisons. D'une part, c'est elle qui permet de transformer les données qualitatives en données quantitatives desquels seront dérivées tous les résultats et analyses de la recherche. De ce fait, un encodage jugé invalide invaliderait à son tour l'ensemble des résultats de l'étude. D'autre part, cette phase est en soit la plus subjective de la recherche étant donné qu'elle repose en majeure partie sur le jugement, les connaissances et l'expérience des codeurs.

Ainsi, la mesure d'accord, comme son nom l'indique, permet d'évaluer l'accord entre les codeurs afin de déterminer si cet accord est attribuable au hasard ou s'il y a, à divers degrés, un réel accord entre les deux. Cela peut être réalisé de deux façons. Premièrement, en évaluant l'accord intra-codeur, c'est-à-dire l'accord entre l'encodage d'un segment de données réalisé par un codeur dans un premier temps et l'encodage du même segment de données réalisé par le même codeur dans un deuxième temps. Deuxièmement, en évaluant l'accord inter-codeurs, c'est-à-dire

l'accord entre l'encodage d'un segment de données réalisé par un premier codeur et l'encodage du même segment de données réalisé par un second codeur.

Par ailleurs, afin de calculer la mesure de l'accord intra et inter-codeurs, la métrique choisie revêt une importance significative. À cet égard, l'emploi d'une métrique inadéquate pourrait remettre en cause la validité des résultats, sans nécessairement les invalider systématiquement, compte tenu qu'il existe bon nombre de méthodes de calcul d'accord. En effet, Popping (Popping, 1988) dénote quelques 39 indices d'accord de catégories nominales différentes dont quelques-uns sont largement utilisés au détriment de d'autres. Ayant tous des hypothèses et propriétés différentes de même qu'un contexte précis dans lequel ils doivent être employés, aucun ne parvient à faire l'unanimité entre les chercheurs. Il convient donc à chaque chercheur, d'une part, de faire un choix parmi cet ensemble d'indices d'accord et en justifier l'emploi, mais d'autre part, il serait également avisé de faire l'usage de plus d'un indice applicable afin de corroborer les résultats exposés.

Or, la première métrique qu'il serait possible d'envisager est la proportion d'accord. Bien que ce calcul est des plus simples à réaliser et semble de prime abord représenter fidèlement le phénomène désiré, il n'en est pas, puisqu'il comporte l'énorme lacune de ne pas tenir compte du fait qu'un même code peut être attribué au même segment de données simplement fortuitement, par le fruit du hasard. De ce fait, plus le nombre de codes possibles est restreint, plus la probabilité de coïncidence due au hasard sera grande. Cela ne semble donc pas être la métrique appropriée.

Une seconde métrique de mesure d'accord envisageable est celle du kappa de Cohen (Cohen, 1960; Cohen, 1968). Ce coefficient où la valeur 1 indique un accord parfait et la valeur 0 indique un désaccord total entre les codeurs, comporte entre autres l'avantage d'éliminer la proportion due au hasard dans la mesure d'accord contrairement à la proportion simple. Le calcul du kappa de Cohen, calculé à partir d'une matrice de contingence n par n dénombrant les occurrences de chaque paire des n catégories, est le suivant :

$$k = \frac{P_o - P_c}{1 - P_c}$$

où

$$P_c = \sum_i P_i \cdot P \cdot i$$

et

$$P_o = \frac{\sum_i a_{ii}}{N}$$

Ainsi, dans les formules ci-dessus, P_o représente la proportion simple d'accord observé alors que P_c indique la proportion d'accord espéré. Toutefois, le calcul de P_c implique que la proportion marginale de chaque catégorie P_i doit être connue d'avance, ce qui cause problème dans ce cas-ci étant donné que cette recherche est exploratoire et que cette proportion est inconnue au moment de l'encodage. De ce fait, étant donné que le hasard est ainsi défini et connu d'avance, si un écart survient entre la proportion marginale espérée et celle observée, le kappa deviendra plus conservateur puisque sa valeur maximale deviendra inférieure à 1. En dépit du fait que cette méthode de calcul est largement utilisée dans la littérature, elle ne peut pas s'appliquer dans le présent cas.

Le kappa proposé par Brennan et Prediger (Brennan & Prediger, 1981) résout toutefois le problème posé par le kappa de Cohen puisque ceux-ci proposent une définition différente du hasard. En fait, au lieu de fixer le hasard à une proportion attendue, le kappa de Brennan et Prediger remplace P_c dans l'équation du kappa de Cohen par $1/n$ suggérant ainsi d'adopter la prémisse de l'équiprobabilité des catégories. Bien que cette définition du hasard ne soit pas tout à fait appropriée à cette étude puisque selon la définition de nos catégories, il est permis de croire qu'elles ne seront pas toutes observées dans les mêmes proportions, elle est néanmoins plus près de nos besoins que celle proposée par Cohen. Ainsi, le calcul du kappa de Brennan et Prediger est le suivant :

$$k = \frac{P_o - 1/n}{1 - 1/n}$$

Une autre métrique bénéficiant d'un relativement large consensus au sein de la communauté de chercheur, quoique moins utilisée que le kappa, est l'index d'accord de Perreault et Leigh (Perreault & Leigh Laurence E., 1989). Contrairement au kappa, cet index ne pose pas l'hypothèse de l'équiprobabilité des catégories et semble donc plus approprié à cette recherche. L'index de Perreault et Leigh s'exprime sous la forme suivante :

$$I_r = \sqrt{((P - 1/n) \cdot (n / (n - 1)))}, \forall P \geq 1 / n$$

Finalement, l'alpha (α) de Krippendorff (Krippendorff, 2004) est également une autre métrique bien acceptée dans la littérature, principalement dans le domaine de

la communication. Ce coefficient très flexible permet de calculer la mesure d'accord à plusieurs codeurs, pour différentes tailles d'échantillons, de même qu'avec des données manquantes. Toutefois, le calcul de Krippendorff est beaucoup plus compliqué et demande, pour ce faire, l'aide d'outil automatisé. Il consiste d'abord à constituer une matrice où chaque colonne représente le code attribué par chaque codeur à une unité de données et qui prend la forme suivante :

Unités u :	1	2	.	.	.	u	.	.	.	r	
Codeurs:	1	c_{11}	c_{12}	.	.	.	c_{1u}	.	.	.	c_{1r}
	i	c_{i1}	c_{i2}	.	.	.	c_{iu}	.	.	.	c_{ir}
	j	c_{j1}	c_{j2}	.	.	.	c_{ju}	.	.	.	c_{jr}

	m	c_{m1}	c_{m2}	.	.	.	c_{mu}	.	.	.	c_{mr}
Nombre de valeurs dans u :	m_1	m_2	.	.	.	m_u	.	.	.	m_r	

Une matrice de coïncidence doit être ensuite formée sous la forme suivante :

Catégories :	1	.	k	.	.	
1	o_{11}	.	o_{1k}	.	.	n_1
.
.
c	o_{c1}	.	o_{ck}	.	.	$n_c = \sum_k o_{ck}$
.
	n_1	.	n_k	.	.	$n = \sum_c \sum_k n_{ck}$

où

$$o_{ck} = \frac{\sum_u \text{nombre de paires } c - k \text{ dans l'unité } u}{m_u - 1}$$

et où $c = k \equiv$ une paire et $k = c \equiv$ une seconde paire.

Finalement, l'alpha (α) nominal se calcule par l'équation suivante :

$$\alpha \text{ nominal} = \frac{(n-1) \sum_c o_{cc} - \sum_c n_c (n-1)}{n(n-1) - \sum_c n_c (n-1)}$$

2.2.6.1.1.1 Résultats

Afin de procéder à la mesure de l'accord, un certain nombre de sessions de travail observées ont été recodées, c'est-à-dire, cinq dans le cas de l'accord intra-codeur et trois pour ce qui est de l'accord inter-codeurs. Ces sessions ont été choisies au moyen d'un échantillonnage aléatoire systématique appliqué depuis la première session de travail encodée et ce, à toutes les trois sessions jusqu'au compte de cinq. Ainsi, la validation se faisait plutôt à partir des premières sessions et cela était souhaité pour diverses raisons. D'abord, dans le contexte du recodage afin de réaliser la mesure de l'accord intra-codeur, cela était souhaitable étant donné qu'approximativement un mois séparait le recodage du premier encodage. Ainsi, il était favorable de choisir les premières sessions afin que le codeur ait le temps d'oublier le plus que possible les séquences du premier encodage et ainsi obtenir des résultats comportant moins de chance de biais. Dans le contexte de l'accord inter-codeurs, il était également préférable de choisir les premières sessions puisque le codeur principal a acquis une certaine expérience tout au long de la première phase d'encodage. Il était ainsi permis de croire que l'expérience de ce codeur était moindre et ainsi davantage comparable à celle du second encodeur lors

de l'encodage des premières sessions de travail, que lors des dernières. Les sessions de travail sélectionnées pour la mesure d'accord sont répertoriées en annexe du présent document.

Par ailleurs, dans le cas de la mesure d'accord intra-codeur, la validation a été réalisée en deux étapes. La première consistait à recoder entièrement et à l'aveugle les deux premières sessions de travail sélectionnées à l'aide de l'échantillonnage expliqué précédemment. Cette étape avait pour but, d'une part, de valider la fiabilité du découpage des données brutes en séquences, tâche équivalente à celle de «segments» telle que définie par Fisher et Sanderson (Sanderson & Fisher, 1994), et d'une autre part, de valider l'encodage des séquences et plus particulièrement la catégorisation des activités collaboratives *ad hoc* observées. Une fois fait, il apparaissait clair que le découpage des séquences constituait une opération très peu subjective puisque seulement 5 séquences ont été découpées différemment sur un total de 94 entre les deux versions. Bien que la même période de temps ait été attribuée à une ou deux séquences données dans les deux versions, les différences entre celles-ci sont explicables par le fait qu'une séquence dans une version ait été découpée en deux séquences dans l'autre version pour y attribuer deux catégories d'activités différentes. Une deuxième étape de validation a donc été réalisée approximativement un mois plus tard, mais cette fois-ci, dans le but de réduire le ratio TA :TS évoqué au cours du chapitre portant sur la démarche d'analyse, le recodage des cinq sessions de travail choisies a été réalisé à partir du découpage en séquences du premier encodage. Ainsi, le codeur n'avait qu'à catégoriser à l'aveugle chaque séquence de données prédécoupée. Ce choix est justifié d'une part, par le fait que la catégorisation (choisir le type d'activité

collaborative *ad hoc* tel que partage d'information, synchronisation cognitive, développement, coordination, etc..) des activités collaboratives *ad hoc* observées constitue une des sources d'information importante de cette recherche et d'une autre part, parce qu'elle constitue un type de données subjectif pouvant laisser place à diverses interprétations. C'est cette façon de procéder lors de cette dernière étape qui fut d'ailleurs répétée lors de la validation inter-codeurs.

Le tableau suivant illustre donc les résultats de mesure d'accord intra et inter-codeurs obtenus dans le cadre de cette recherche pour les métriques retenues.

Tableau 2.1 Résultats de la mesure d'accord intra et inter-codeurs en fonction des métriques retenues

	Proportion simple	Kappa de Brennan et Prediger	Index de Perreault et Leigh	Alpha (α) de Krippendorff
Accord intra-codeur (étape 1) :	0.82	0.79	0.89	0.80
Accord intra-codeur (étape 2) :	0.98	0.97	0.99	0.97
Accord inter-codeurs :	0.66	0.62	0.79	0.57

L'analyse du tableau ci-dessus démontre dans un premier temps que l'accord n'est pas parfait tant pour l'accord intra-codeur que l'accord inter-codeurs, mais les résultats sont néanmoins intéressants. D'abord, il apparaît qu'en règle générale, l'accord intra-codeur semble beaucoup plus élevé que l'accord inter-codeurs. Cela va dans le même sens que ce à quoi il était en droit de s'attendre étant donné qu'il est plutôt logique qu'un codeur soit davantage constant avec lui-même, qu'avec un autre codeur. Du côté de l'accord intra-codeur, les résultats obtenus semblent démontrer un accord important du codeur avec lui-même, voire excellent dans le cas de la seconde étape. Dans ce dernier cas, sur les 227 séquences de données à recoder, cinq désaccords ont pu être observés entre les deux versions. Cela est probablement attribuable à l'expérience acquise par ce codeur après plusieurs dizaines d'heures d'encodage. Ainsi, les proportions simples indiquent respectivement 0.82 et 0.98 entre les deux versions ce qui suggère non seulement un accord significatif avec la version de référence, mais également une amélioration de l'accord avec le temps supportant l'idée du gain d'expérience. Le kappa de Brennan et Prediger affiche quant à lui des résultats un peu plus faibles par rapport aux proportions simples étant donné que la part attribuable au hasard a été éliminée du calcul d'accord. Toutefois, les résultats demeurent toujours très importants. D'autre part, l'analyse de l'index de Perreault et Leigh démontre également un fort accord suggérant même un accord plus important que le kappa de Brennan et Prediger. Quant à l'alpha de Krippendorff, il démontre lui aussi un accord important puisque dans le cas de la première étape, il indique un accord de 0.8, tout juste sur la limite de 0.8 fixée par l'auteur pour indiquer un accord pouvant mener à des conclusions définitives (Krippendorff, 2004), et dans le cas de la seconde étape, tel que mentionné précédemment, l'accord semble quasi-

indéniable. Bien que ces données soient encourageantes puisqu'elles témoignent d'une certaine constance et cohérence du codeur principal avec lui-même, il serait toutefois intéressant d'explorer un peu plus en détail les résultats obtenus pour l'accord inter-codeurs et de savoir si la même cohérence peut être observée entre deux codeurs différents.

Ainsi, la lecture de la proportion simple du côté de l'accord inter-codeurs démontre un accord qui semble beaucoup moins fort que l'accord intra-codeur. Le kappa de Brennan et Prediger, comme on peut s'y attendre corrobore cette première observation en retranchant la part de hasard de la proportion simple. Ces résultats ne permettent donc pas de conclure à un accord inter-codeurs important s'ils sont comparés au consensus généralement accepté dans la littérature à l'effet qu'un indice d'accord de kappa doit être supérieur ou égal à 0.7 afin de tirer pareille conclusion. Toutefois, l'analyse de l'index de Perreault et Leigh démontre des résultats beaucoup plus importants puisque l'index obtenu se situe à 0.79, soit clairement au-dessus de la barre de 0.7 fixée par ces auteurs pour statuer d'un accord important. Un tel gain par rapport au résultat obtenu au kappa de Brennan et Prediger peut potentiellement s'expliquer par l'hypothèse de l'équiprobabilité des catégories posée par ces derniers. En effet, les résultats d'analyse préliminaires communiqués par Cherry et Robillard (Cherry & Robillard, 2004b; Cherry & Robillard, 2004a) et consolidés par les résultats affichés au cours des prochains chapitres, démontrent que la distribution des catégories est fortement inégale au point où elle peut très bien influencer les résultats obtenus compte tenu du nombre relativement restreint de catégories. Toutefois, l'alpha de Krippendorff observé vient un tant soit peu atténuer le très bon résultat obtenu à l'index de Perreault et Leigh

puisque'il se situe à 0.57 ce qui est sous la barre de 0.67 proposée par l'auteur pour invoquer des résultats potentiellement concluants par opposition à l'index de Perreault et Leigh qui permettait de prétendre à un accord important. Bien que ce résultat surprenant vienne appuyer le fait que chaque indice d'accord soit basé sur des hypothèses différentes et qu'il est essentiel d'en faire usage de quelques-uns afin de faire corroborer les résultats entre eux, il serait néanmoins intéressant de chercher à comprendre pourquoi une telle dissemblance entre l'index de Perreault et Leigh et l'alpha de Krippendorff a pu être obtenue.

Or, il semblerait que cette différence soit attribuable au conservatisme du deuxième codeur puisque'une simple analyse des divergences entre le premier et le second codeur démontre que bon nombre de celles-ci se sont produites lorsque ce dernier s'est abstenu de catégoriser l'activité donnée et ait plutôt opté pour la catégorie non-identifiée. Cette catégorie a été ajoutée aux autres catégories présentées plus tôt afin de pouvoir éliminer les séquences qui étaient très difficiles à coder pour diverses raisons, que ce soit parce que par le verbalisme il était impossible de le faire ou parce que la qualité sonore ne le permettait pas. De plus, cette catégorie échappatoire permettait de faire diminuer la part de subjectivité associée au codage de cet attribut de donnée étant donné que l'attribution d'une catégorie par la chance était ainsi proscrite. Ainsi, la mesure de l'accord inter-codeurs sans tenir compte des séquences où au moins un des deux codeurs ne pouvait pas se prononcer donne les résultats présentés au tableau suivant :

Tableau 2.2 Résultats de la mesure d'accord inter-codeurs en fonction des métriques retenues et pour les activités identifiées par les deux codeurs

	Proportion simple	Kappa de Brennan et Prediger	Index de Perreault et Leigh	Alpha (α) de Krippendorff
Accord inter-codeurs :	0.81	0.79	0.89	0.76

Ces résultats démontrent un accord nettement plus important que les résultats tenant compte des séquences non-identifiées. La proportion simple, le kappa de Brennan et Prediger ainsi que l'index de Perreault et Leigh obtenus démontrent entre eux la même tendance observée par les résultats précédents, mais ils affichent également tous une valeur nettement au delà de 0.7 pouvant ainsi suggérer un accord important entre les codeurs lorsque ceux-ci étaient tous les deux en mesure de se prononcer. Ces résultats sont également supportés par l'alpha de Krippendorff qui indique une valeur largement au-dessus de la barre du 0.67 qui permet d'énoncer des tentatives de conclusions, voire même un peu en deçà de la barre du 0.8 permettant d'émettre des conclusions plus arrêtées.

2.2.6.2 Fiabilité

La fiabilité de la méthode ou d'une mesure, réfère à la capacité de reproduire les mêmes résultats dans les mêmes conditions. Bien que la fiabilité constitue un aspect primordial dans le cadre de recherches expérimentales afin que les résultats puissent jouir d'une certaine crédibilité, dans le cas de cette étude-ci, et dans un sens plus large, dans le cadre d'études de cas et d'observations participantes, la

fiabilité à moins de sens puisque chaque méthode de collecte de données est adaptée au contexte du terrain et cela rend ainsi difficile la répétition de cette dernière (Jorgensen, 1989). Par conséquent, les techniques de collectes de données utilisées au cours de ce projet sont fortement adaptées au contexte du terrain sur lequel s'effectue la recherche ce qui rend donc la reproductibilité de la technique de mesure plus difficile. Elles ont néanmoins été décrites en détail au cours de ce chapitre, facilitant la reproductibilité de la méthode adoptée dans le cadre de contexte semblable à cette étude et conférant également un certain niveau de qualité à la recherche présentée dans le présent document.

Toutefois, étant donné que la fiabilité est plutôt liée à la solidité et la véracité des découvertes lors de ce type de recherche (Jorgensen, 1989), et que cela est intimement lié à la validité interne de la recherche tel qu'il fut discuté précédemment, il serait très appréciable que les résultats issus de cette étude puissent être transférables d'un contexte ou d'un terrain à un autre pour que les théories proposées ainsi que les retombées pratiques qui y sont dérivées soient applicables en dehors du cas spécifique étudié.

CHAPITRE 3

PRÉSENTATION DES ARTICLES

Ce document, développé sous forme de thèse par articles, constitue le point culminant d'un important travail de recherche qui a permis l'approfondissement des connaissances, tant au point de vue de l'objet à l'étude en lui-même, que des méthodes qui ont permises de l'étudier. Les résultats des travaux réalisés de même que les analyses pertinentes sont donc rapportés au cours des trois chapitres suivants sous forme d'articles rédigés, soumis et, publiés pour certains, à des revues avec comité de lecture, constituant ainsi le corps du présent document. Parmi ceux-ci, divers articles de conférences ont été rédigés et présentés afin de dévoiler rapidement des résultats de cette étude, mais également afin d'obtenir l'impression et des commentaires de la part de chercheurs du domaine. Ces articles de conférences sont intégrés en annexe du présent document et sont relatées séquentiellement ci-dessous afin de comprendre l'évolution progressive et itérative des travaux d'analyse du travail collaboratif *ad hoc* étudié. Il n'est toutefois pas nécessaire pour le lecteur de lire les articles de conférences en annexe, tous les résultats pertinents de la recherche ayant été rapportés au sein des articles de revue faisant l'objet des prochains chapitres.

D'abord, deux premiers articles de conférences ont été présentés en 2004. Alors que le premier des articles intitulé «Empirical Study of Ad Hoc Collaborative Activities in Software Engineering» présenté dans le cadre du *First International Workshop on Computer Supported Activity Coordination* tenu en marge de

l'International Conference on Enterprise Information Systems (ICEIS 2004) à Porto au Portugal visait, non seulement à présenter le sujet de recherche et en quoi celui-ci constituait une piste d'amélioration de processus intéressante, il visait également à présenter la méthodologie de recherche employée pour ce faire, de même que les premiers résultats préliminaires obtenus des premières analyses effectuées. Le second article, intitulé «Communication Problems in Global Software Development: Spotlight on a New Field of Investigation» cette fois présenté un mois plus tard dans le cadre du *Third International Workshop on Global Software Development (GSD 2004)* en marge de *l'International Conference on Software Engineering (ICSE 2004)* à Édimbourg en Écosse visait à cadrer la pertinence du sujet de recherche dans le domaine du *Global Software Development* qui constitue une sphère de recherches de plus en plus d'intérêt étant donné l'augmentation du recours à de telles pratiques dans l'industrie.

Un troisième article de conférence fût présenté dans le cadre de la conférence *Ergo'IA 2006 - L'humain comme facteur de performance des systèmes complexes* à Bidart-Biarritz en France et s'intitulait «Importance of Peer-to-Peer Ad hoc Collaboration in the Development of Large Software Systems». Non seulement celui-ci permit de réaffirmer l'importance du sujet de recherche et d'obtenir des commentaires constructifs de la part des réviseurs anonymes de même que de l'assistance, il permit également de présenter, à la suite de plus amples analyses, quelques premiers patrons quant aux acteurs impliqués et aux rôles occupés au sein du travail collaboratif *ad hoc* de même que les mécanismes d'échanges privilégiés. Cet article fût d'ailleurs instigateur du premier article de revue formant le corps de cette thèse.

Ainsi, le chapitre 4 présente le premier des trois articles de revue intitulé «The Social Side of Software Engineering – A Real *Ad hoc* Collaboration Network», paru en juillet 2008 au sein d’une édition spéciale titrée *Collaborative and social aspects of software development* de la revue *International Journal of Human Computer Studies*. L’article traite donc des divers aspects de la recherche en les abordant d’un point de vue sociologique. En premier lieu, les motivations quant à l’étude du travail collaboratif *ad hoc* sont présentées et placés dans un contexte de mesure des aspects humains du génie logiciel. Par la suite, la méthodologie employée est longuement décrite, elle qui fut fortement inspirée des méthodologies communément utilisées dans les sciences humaines telles que la sociologie et l’anthropologie pour étudier ce sujet d’intérêt relativement nouveau en génie logiciel qu’est l’aspect humain du génie logiciel. Les considérations en ce qui a trait à certains aspects de la méthodologie tels que la méthode de collecte des données, l’échantillonnage des données de même que les aspects éthiques de la recherche sont également discutés, toujours dans un cadre d’étude sociologique de la collaboration dans un environnement de génie logiciel. Enfin, des résultats d’analyses sont divulgués, résultats dévoilant un véritable réseau de collaboration *ad hoc* au sein duquel les acteurs observés occupent chacun un rôle distinctement bien différent les uns des autres et démontrant mieux la part inévitablement sociale du génie logiciel.

Entre-temps, un quatrième article de conférence fût présenté dans le cadre d’un workshop nommé *Cooperative and Human Aspects of Software Engineering (CHASE)* en marge de l’*International Conference on Software Engineering (ICSE)*

2009) tenu à Vancouver au Canada. Cet article intitulé «Audio-Video Recording of ad hoc Software Development Team Interactions» visait à pousser encore plus loin l'analyse des rôles formel et informel occupés au sein d'une équipe de développement ainsi que des canaux et modes de communications employés. À la suite de cette présentation, une invitation à soumettre un article dans la revue *IEEE Software* fût proposée de laquelle l'article présenté au chapitre 5 découle.

Le chapitre 5 expose ainsi le second article de la série d'articles de revue, intitulé «*Ad hoc* Collaboration in a Software Development Setting – Why and How? », soumis pour considération dans la revue *IEEE Software*. Outre quelques précisions méthodologiques pertinentes par rapport à l'article de revue précédent, notamment quant au modèle de données développé, cet article vise principalement à fournir les premiers éléments de réponses à savoir *pourquoi* et *comment* est réalisé le travail collaboratif *ad hoc*. Une analyse du point de vue des différents acteurs, introduits lors de l'article précédemment publié, est également développée, démontrant les patrons comportementaux émergents quant aux canaux de communication utilisés et mais également à l'égard des types d'activités collaboratives *ad hoc* dans lesquels les acteurs prennent part à ce qui constitue en soit une première communication de résultats, mis à part les résultats préliminaires divulgués en 2004, quant au contenu du travail collaboratif *ad hoc*.

Finalement, le chapitre 6 présente le dernier article, intitulé «Empirical Analysis of *Ad hoc* Collaboration Patterns and Content in a Co-Located Software Development Environment – A Case Study» soumis pour la revue *IEEE Transaction on Software Engineering*. Cet article beaucoup plus substantiel complète la boucle de la série

d'articles en présentant, d'abord, des éléments méthodologiques pertinents pour enfin porter plus d'emphasis sur des résultats d'analyses approfondis quant aux patrons et au contenu analysés des activités collaboratives *ad hoc* observées rendant possible une meilleure compréhension de la nature opportuniste de ce type d'activités et de la teneur des échanges collaboratifs au sein d'une équipe de concepteurs de logiciels. Cette meilleure compréhension permettant l'élaboration de pratiques ayant comme objectif premier d'améliorer la collaboration dans un environnement de développement de logiciels, un certain nombre de telles pratiques ont été proposées rendant ainsi possible leur validation par de futures recherches qui viseront à poursuivre les efforts entrepris dans le cadre de la présente étude et pour lesquelles, quelles pistes éventuelles ont été fournies.

CHAPITRE 4

THE SOCIAL SIDE OF SOFTWARE ENGINEERING – A REAL AD HOC COLLABORATION NETWORK

Sébastien Cherry, Pierre N. Robillard

Software Engineering Research Laboratory, Computer Engineering Department,
École Polytechnique de Montréal, C.P. 6079. succ. Centre-Ville, Montréal, Canada

Phone (514) 340-4711 ext. 4238. Fax (514) 340-5139

{Sebastien.Cherry, Pierre-N.Robillard} @ polymtl.ca

4.1 Abstract

Recognized first and foremost as a technical task, the construction of software is, for the most part, a human experience. Software development is an intensive cognitive task which also constitutes "an exercise in complex interrelationships," according to Fred Brooks (Brooks, 1975). This paper presents a case study conducted to analyse ad hoc collaborative activities taking place between team-mates during an industrial software development cycle. Observations based on audio-video recording are analysed with a methodology from social sciences research. The paper describes the observational approaches, the various methods used to validate data and how quantitative data are extracted from the qualitative observations of team-mates. Patterns of ad hoc collaboration emerging from this study are revealed, which lead us to believe that both the implicit and explicit roles of partners have a definitive impact on their ad hoc interaction profiles. As a result, this study helps provide an understanding of some of the pivotal aspects of software

engineering, such as collaboration, coordination and, more generally, work team dynamics. Further details relating to our initial motivations are included, followed by a comprehensive description of the methodological approach designed specifically for this research. Finally, some results are presented, which shed light on a real ad hoc collaboration network and support the importance of the human and social aspects of software engineering in a more substantial way.

4.2 Introduction

Software engineering is becoming increasingly recognized as a human experience, although the human aspects involved in software development have been studied little in the past. This gap in the research is, according to Perry, Staudenmayer and Votta (Perry et al., 1994-), primarily due to the difficulty in quantitatively measuring the human facets of engineering. However, as observed by Seaman (Seaman, 1999-), more and more studies have been aimed at measuring the “people factors”. Among them, communication (Seaman & Basili, 1997), coordination (Grinter et al., 1999; Herbsleb & Grinter, 1999) and collaboration (Robillard & Robillard, 2000) (Potts & Catledge, 1996) between software engineers are some of the relevant aspects on which researchers have focused in recent years.

It has been observed that, team members frequently interact on an *ad hoc* basis during a software development project. *Ad hoc* activities are defined as interactions forming a logical communicative unit containing one or more sequences with obvious internal continuity, while remaining distinct from what precedes it or what follows it (Kerbrat-Orecchioni, 1998). Simply put, *ad hoc* communication arises

when one team member spontaneously interrupts a team-mate to ask him a question or to provide unsolicited information; for instance: "Hey Martin, could you please come here and look at this dump?" Martin will then move to his colleague's desk and the two will interact for a while. The duration of the interaction is unpredictable and can last from a few seconds to many minutes. In this typical example, the *ad hoc* interaction ended when the conversation between Martin and his colleague ceased. *Ad hoc* collaboration occurs mostly through face-to-face communication. Traditional disciplined software engineering processes have no place for such *ad hoc* communications in their prescribed practices. Agile approaches, by contrast, promote verbal exchanges in lieu of documentation, which is more like a scheduled meeting where the teammates agree to work together. *Ad hoc* collaborations are very time-consuming and the content and usefulness of the communications are rarely documented.

Ad hoc collaboration is a natural human behavior observed in any kind of teamwork. The overall objective of this study is to better understand this behavior. Our first specific objective is to quantify the phenomena and to find out how *ad hoc* communications are related to the role team-mates play.

Following along the lines of the studies mentioned above, Perry, Staudenmayer and Votta, who are among the few pioneers in the field, have tried to describe how developers spend their time as a software development project unfolds (Perry et al., 1994-). Their studies show, among other things, that developers spend 75 minutes per day on unplanned informal communications. This observation was later supported by Robillard and Robillard in their study on the different types of

collaborative work in software engineering (Robillard & Robillard, 2000). The types of work they identified are mandatory meetings, called or scheduled meetings, *ad hoc* meetings and individual work. A later study by d'Astous and Robillard (d'Astous & Robillard, 2002a) analysed the patterns of exchange between team-mates during called meetings and during meetings scheduled in advance such as peer-review meetings. For this, they built a model to represent the qualitative and quantitative importance of various exchanges occurring during peer-review meetings. Like *ad hoc* meetings, they are more difficult to study given their spontaneous nature. They are nevertheless important, since they have been observed by Robillard and Robillard to account for a quite a large proportion of the time spent on a software project, as also observed by Perry and his colleagues, who quantified it at 41%. These observations convinced them of the importance of these activities to team dynamics, a conviction shared by many more authors in subsequent studies, e.g. Herbsleb and Grinter (Herbsleb & Grinter, 1999) and Seaman and Basili (Seaman & Basili, 1997). However, despite this fairly large consensus, no known study has yet described and characterised *ad hoc* collaborative activities and their communication patterns. This is what we endeavoured to do in the research described in this paper.

4.3 Research Methodology

Although the theoretical foundations exist for studying human behaviour in working environments in various domains, such as communication psychology or organizational sociology, not much has been written on *ad hoc* collaborative work. Consequently, we were forced to conduct this research in exploratory mode. Unlike confirmatory studies, which are aimed at validating a hypothesis postulated

beforehand, with the final objective of confirming or rejecting the validity of theories which have already been defined, exploratory studies generate new theories on little-known fields, which will later be validated by confirmatory studies. Exploratory studies also typically focus on small samples and case studies by means of in-depth methods such as open interviews, focus groups or participant observations. The participant-observation method, which is broadly employed in sociology and anthropology for the study of individuals or groups of individuals, is well suited in the following cases (Babbie, 2001), when (Jorgensen, 1989):

- observing the events and phenomena under study in real time and collecting the associated data;
- studying the behaviour and interpersonal relationships of participants in their natural environment;
- conducting exploratory and descriptive research;
- the phenomena under study are observable in everyday life and cannot be reproduced in an experimental context;
- the phenomena under study are limited enough to be observed in the form of a case study;

The participant-observation method is therefore particularly well suited for exploratory and descriptive studies such as this one. Problem definition using this method is, according to Jorgensen (Jorgensen, 1989), the result of an open and flexible process during which what will be studied is identified, clarified, refined and elaborated during the course of the study. In addition, concepts used for the observation and analysis of the data are defined from the meanings derived from

the field context. This means that defining or operationalising concepts in advance, as is done in confirmatory studies, can contaminate participant observers and feed them a preconceived idea of what might be found. In that sense, it can very considerably distort their point of view and lead them to erroneous or biased conclusions.

4.3.1 Methodological Traditions

Sanderson and Fisher have described three approaches to the study of human behavior in the methodological tradition (Sanderson & Fisher, 1994). The *behavioural tradition* is concerned with human behaviour in its natural environment. Studies in this tradition tend to be more objective and quantitative, and rely on sampling theories for subjects and data sequence selection. The encoding will be formalized using standardized terms with a focus on the reliability and reproducibility of results under the same controlled conditions. The *cognitive tradition* focuses on human cognitive performance in the accomplishment of particular tasks. Studies in the cognitive tradition tend to be explanatory and typically attempt to explain theories which are often generated by behaviourists. Thus, instead of working with a large sample as behaviourists do, cognitivists are more likely to work with very small samples, but do so more deeply and often in laboratory settings. Studies in the *social tradition* focus on the interactions and communications of individuals in their environment, i.e. human-human or human-computer. This tradition favours field observations and values direct observer participation in an effort to cover representative situations, instead of relying on a statistical sampling. Investigators do not try to eliminate subjectivity, but rather use

it and explore it by involving many people in the analysis, even the subjects themselves, and do so with the avowed intention of coming up with highly valid findings.

Our methodology is clearly inspired by the social tradition. Our subjects were sampled using a judged sampling method, so that we would analyse subjects who were more likely to provide us with relevant material to analyse. Also, our encoding involved interpretation by the coder, which introduced a degree of subjectivity. This was unavoidable, since observed *ad hoc* collaborative activities have to be well understood in order to identify and categorize their content. Formal encoding methods would have been inadequate for classifying logically observed sequences.

4.3.2 Observer or Participant

The participant-observation method raises an important question, which is the extent to which the observer himself participates in the group of individuals being observed, a continuum stretching from the complete observer, where the participant takes part in all the activities of the group wholeheartedly and even “becomes” a full-fledged member of that group, to the complete participant, where the observer does not take part in the activities of the group at all and remains on the fringe. Between these two end-points, the observer can be an observer-participant, acting more as an observer than as a participant, and vice-versa. Generally speaking, and this point has also been brought up by Jorgensen (Jorgensen, 1989), the roles of *participant* and *observer* are often seen to be in contradiction: when an individual participates more actively, he will be less prone to

observing, while, conversely, more observation will impede participation. This raises the issue of objectivity vs subjectivity in qualitative research, but the dichotomy lost some of its relevance when a consensus emerged among participant observers that, in order to avoid misunderstanding and misinterpretation of observations, the observer must get closer, both physically and socially, to those being observed, who are then called "insiders". In other words, as suggested by Lofland and Lofland (Lofland & Lofland, 1995), and cited by Babbie (Babbie, 2001), the observer must gain access to "insider understanding", i.e. the subjective reality in which the so-called insiders live. This is who Jorgensen (Jorgensen, 1989) refers to as the participant observer who willingly "becomes" part of the phenomenon in order to extract accurate data and reach accurate conclusions. However, joining the phenomenon, and thus becoming part of the object under study, will make it more difficult for the observer to take a step back to reassess his research framework and plot a course of action adapted to new circumstances.

One of the main advantages of the participant-observation method is that it can generate highly valid results, since the definition of the underlying concepts is inspired directly from the reality of those being observed (whom we refer to as *observees*). However, the reliability of the research is less relevant in participant observation because the collection methods used are usually adapted to the field context, which makes reproducibility more difficult. Nevertheless, reliability in participant observation could be related to the solidity and veracity of the discoveries (Jorgensen, 1989).

During the course of this research, one of the aspects of which is reported in this paper, we decided that the observer should act as a complete participant for several reasons, the most important being that the aim of the research was to study *ad hoc* collaborative interactions, including their content. The nature of the interactions between the software developers observed required that the observer be in a position to understand the jargon used by the observees in their everyday work, as well as the culture of the organisation and the sub-culture of the team. Thus, in order to reach this level of understanding, the observer had to become fairly actively involved in the activities of the group. Moreover, in order to observe situations which were as valid as possible, the observer had to adopt the standpoint of an insider and thus, as discussed above, become an insider himself. However, to avoid the introduction of subjectivity, the data were also validated and analysed by researchers who were not involved on-site.

4.3.3 Industrial Environment Observed

To record our observations, a software development team from a large software development organisation was approached and they agreed to participate in our research. This team was composed of 12 individuals (i.e. 1 project manager supported by 11 software developers) ranging widely in age, with different levels of schooling (from a Bachelor's degree to a Ph.D. in computer sciences and engineering) and individual experience ranging from 2 to 16 years in the field and between 9 months and 5 years of service in the company. In addition, this team is evolving within a large software organisation which has a highly standardised process comprising several thousand software developers in several countries

around the globe. In spite of the size of the organisation, the setting has the attributes of smaller organisations, as development is shared among several small teams of up to 15 members each, often located at a single site. Having said that, developers regularly need to cooperate with stakeholders in remote locations and in different time zones, because they are divided by functional expertise and are also segregated by product area or software component, both of which are tightly integrated to form an enormous solution of millions lines of code from a number of different technologies. In this complex context, collaboration with a wide range of experts becomes essential.

4.3.4 Observation Phases

The observations were gathered in two phases. The first, which we called the *ethnographic phase*, spanned a several months, during which time the observer was integrated into the group as a regular employee, i.e. as a software developer in the team investigated. Not only did this period allow the observer to become familiar with the organisation, its specific culture and the jargon used by its members, but it also allowed him to plan the appropriate data collection method to be applied during the second period, which lasted two months.

The data collection method used during the observation phase was carefully designed. Often the favoured method used in participant observation is to note down the observations. This can sometimes take place in an informal manner. Data collection can also take place more formally, by means of a grid, for instance. Because of the specific nature of our study, we believed that collecting the data by

audio-visually recording the working sessions would be the most suitable method. We selected it for many reasons, among them the fact that our research was exploratory and we wanted to measure *ad hoc* behaviour, which is unpredictable by nature. This means that our observations would have been difficult to record by simply writing them down, since notes are typically taken when the observer knows what to look for. There were other advantages to this method, one of which is that the tapes can be reviewed repeatedly in order to define and refine the specific attributes to look for in our raw data. This was helpful in our case, as several iterations were needed to define our data model. Another was that, unlike notes, recordings collect detailed information in ways that enable the analyst to move beyond real-time bias.

4.3.5 Data Recording

The issue of potential bias due to the presence of the camera was addressed by starting recording three weeks before our desired date in order to accustom our subjects, who were not aware of this buffer period, to the presence of the camera, thus reducing bias to a minimum. It is worth noting that one subject even admitted to having forgotten the presence of the camera after a couple of days and preferred this to having an observer sit next to him and take notes on his everyday activities. An indication that the behaviour of the observees was not affected by the ever-present camera was that the subjects rarely looked in the direction of the camera once they had become used to it. During the two months of the observation and data collection phase, almost 200 hours of audio-video recordings of working sessions were gathered.

4.3.6 Ethical Issues

Ethical considerations were an important issue, and one which was handled very conscientiously by the participating organisation and the École Polytechnique de Montréal. Every means possible was employed to guarantee the ethical use of the recordings. Our research protocol was approved by an independent committee of the École Polytechnique de Montréal, which is mandated to supervise research undertaken with human subjects, and they issued us with an Ethics Certificate for this research. All the subjects involved in our study were duly informed that their work sessions would be audio-visually recorded, as well as of the nature of the study. They all signed the letter of agreement required for certification. They were told that they were completely free to end their participation in the research at any time. Ethical issues were handled according to the established Canadian policies for research involving humans (NSERC 2005)

4.3.7 Sampling and coding

Detailed observations were made of four of the twelve members on the development team. These subjects were chosen through a judged sampling during our ethnographic phase, because they presented more team behaviours than the others within the context of the larger development team. They were closely co-located, their tasks were related to the same projects, and they were used to working together. They had easy-going characters, a friendly and professional attitude and no personal animosity. This choice is justified because, as we explained in describing the *social tradition*, researchers adopting this approach tend to favour

a realistic and representative sampling instead of relying on statistical sampling. In this way, we arrived at a stratified sampling of working sessions. From this first sample, a number of the analysed working sessions were removed in order to discard bias due to non-representative sessions where some of the subjects were absent or involved in formal meetings, preventing them from collaborating with their colleagues in the usual way. In accordance with the social tradition approach, we analysed a typical working session, which is defined as a session where all four observed subjects are present and which lasts half a day. These sessions are not disrupted by any occasional events, such as visitors, meetings or special duties to be performed by one of the team members.

Finally, around 35 hours of audio-video recordings were analyzed, in which a total of 431 face-to-face *ad hoc* collaborative interactions were observed involving the four subjects interacting with each other, as well as with other stakeholders.

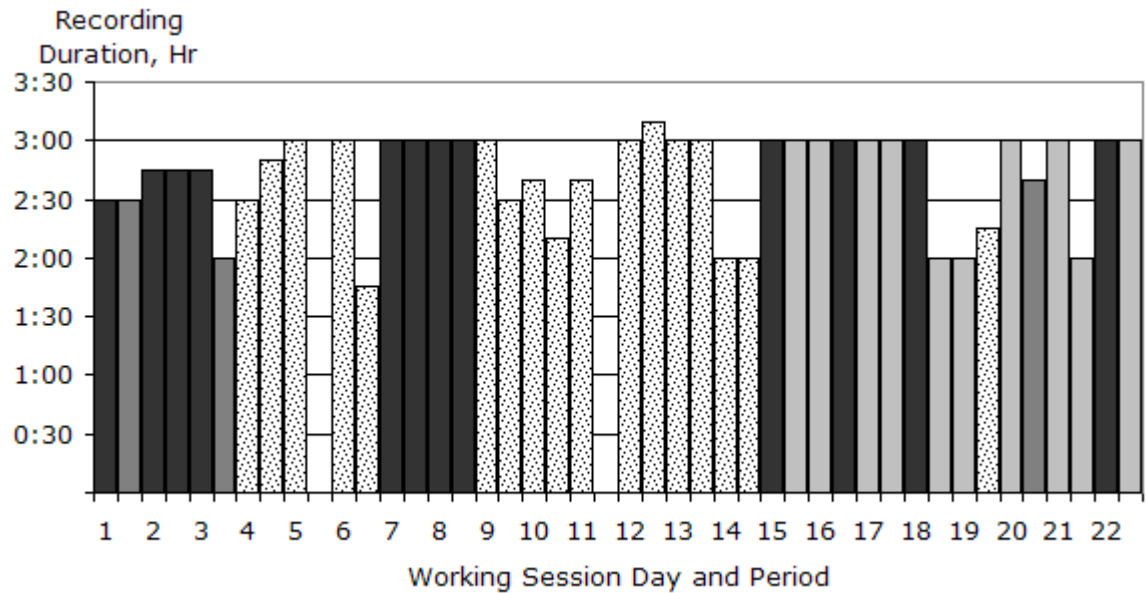


Figure 4.1 Recording duration and sampling per working session day and period

Figure 4.1 shows the duration of the 42 sessions recorded over 22 days at a rate of 2 periods per day (AM: numbered column and PM: unnumbered column). Working session numbers 5 PM and 11 PM were not recorded. The recording was continuous during a session and the recording sessions lasted 2.5 hours on average. The 12 sessions used in this study are highlighted in a darker colour, while the three sessions in medium gray (1PM, 3PM, 20PM) are those that were discarded after analysis because they had been judged to be non-representative; for instance, because some participants' time was monopolised by other meetings for long periods of time. The other sessions available for analysis and discarded by the stratified sampling are coloured in light gray (15PM, 16PM, 17AM-PM, 18PM, 19AM, 20AM, 21AM_PM, 22PM), and the recordings coloured in white are those that were excluded from the start due to the absence of at least one participant.

The granularity we chose for the unit of analysis of this research was the sequence, which is a conversational unit composing the interactions, i.e. what Kerbrat-Orecchioni defines as “a block of one or more exchanges connected by a strong degree of semantic or pragmatic coherence, i.e. dealing with the same topic or centered on the same task” (Kerbrat-Orecchioni, 1998).

4.4 Role Definition

The observations of the team-mates at the start of the study have enabled us to identify the role of each subject within the team dynamics. These roles are often informal and implicit, and not formally acknowledged by the main players themselves. These roles can be better observed during *ad hoc* activities than during formal activities, where organisational and social conventions tend to predominate. Based on qualitative observations, the roles of the four subjects identified as follows:

- The rookie (code name MS1) is the new recruit on the team. Not far below the top of the learning curve, he tries to gather, consolidate and crystallise all the pieces of information needed for an understanding of the environment and of the technical content involved in developing and maintaining software.
- The coach (code name MS2) is the project manager who occupies the formal leadership position. His management style could be described as the intermediary management style defined by Blake and Mouton (Blake Robert

R. & Mouton Jane Srygley, 1964), since his interests are focused as much on the task as on the people involved, hoping for little more than adequate output while he tries to convince and motivate his subordinates by making occasional compromises. He does not hesitate to share relevant information as soon as he receives it, which happens often, as he is the primary interface with external stakeholders.

- The reference (code name MS3) is responsible for configuration management of the software built by the team. He is not to be considered a guru, but he is a fount of knowledge for his colleagues who often ask him for information or help in solving their problems. He holds no formal leadership position like the manager's, and his leadership is informal, rooted mostly in his knowledge and expertise for which he is renowned.
- The cooperator (code name MS4) does not occupy any specific role, formal or informal, on the team. He can be seen as the embodiment of an average developer, performing his tasks well and cooperating fruitfully with his colleagues and other stakeholders, as often as the situation requires.

4.5 Measurement of *Ad hoc* Collaborative Activities

As stated in the introduction, *ad hoc* activities have been identified as interactions forming a logical communicative unit composed of one or more sequences which has obvious internal continuity, while being distinct from what precedes it or follows it, as defined by Kerbrat-Orecchioni (Kerbrat-Orecchioni, 1998). Figure 4.2 shows

the percentage of *ad hoc* collaborative work observed in terms of time spent during a session and for each of the 12 recorded sessions.

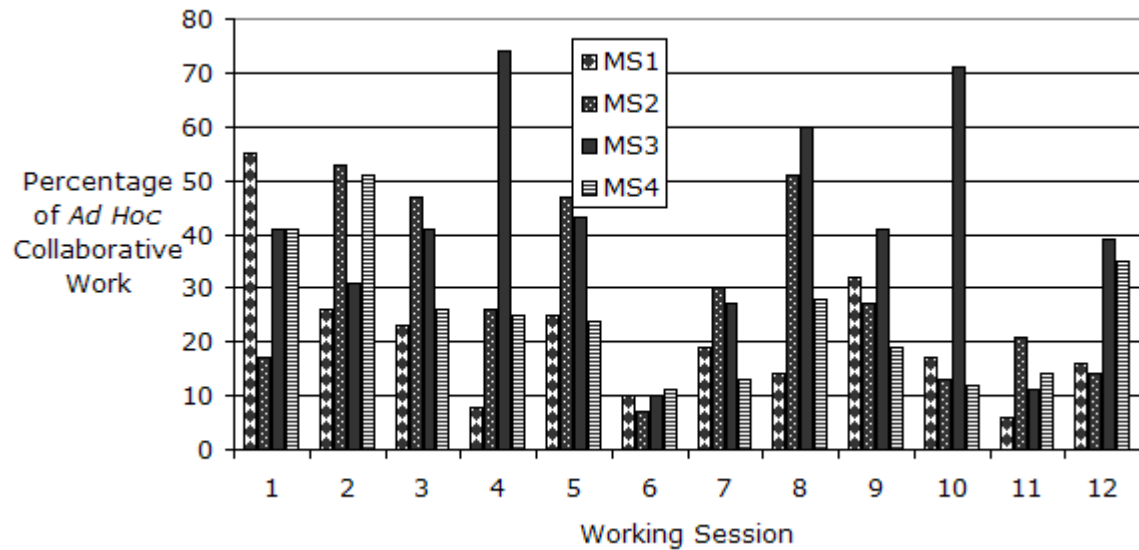


Figure 4.2 Percentage of time spent on collaborative activities per subject for each recorded session

Figure 4.2 illustrates importance of *ad hoc* interactions and the large variations in *ad hoc* activities among sessions. We may recall that *ad hoc* interactions are spontaneous, unplanned, and of an unknown duration. *Ad hoc* activities may be seen by most of the subjects as an interruption of what they are doing. In Figure 4.2, some sessions like 6 and 11, involve little activity, while others (2 and 8) involve a great deal. Sometimes a specific role assumes most of the *ad hoc* activity (4 and 10). This distribution suggests the spontaneous nature of the *ad hoc* activity.

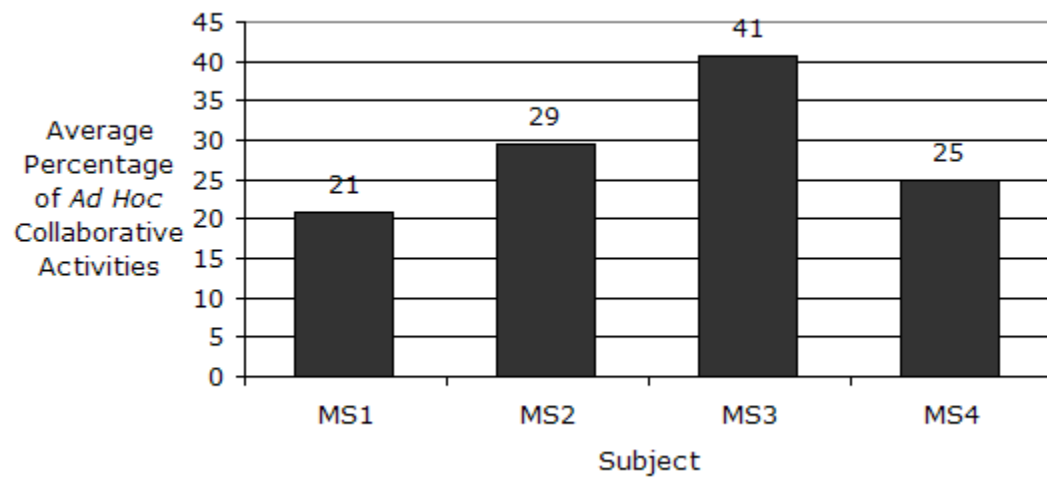


Figure 4.3 Average percentages of *ad hoc* collaborative work observed per subject

Figure 4.3 shows the average percentage of *ad hoc* collaborative interactions over all the sessions for each observee, based on information obtained from Figure 4.2, by averaging the activity of each subject over all the sessions. The two subjects who exhibit one form or another of leadership (MS2 and MS3) spend more time on *ad hoc* collaboration than the other two subjects. While the *coach* (MS2) spent 29% of his total time on *ad hoc* collaborative activities, the reference (MS3) spent as much as 41% of his time on these activities. As for the rookie MS1 and the cooperator MS4, they spent an average of 21% and 25 %, of their time, respectively, on these activities. On a team basis, their involvement accounts for almost 30% of all the activities. We may recall that *ad hoc* interactions are only one component of collaborative activities, however. A team member takes part in three other types of activities: scheduled meetings, mandatory meetings and individual work.

Figure 4.4 shows the average frequency of *ad hoc* collaborative interactions observed per hour per subject. There are from 3 to 5 *ad hoc* interactions per hour.

The coach (MS2) and the reference (MS3) are involved the most frequently in *ad hoc* interactions.

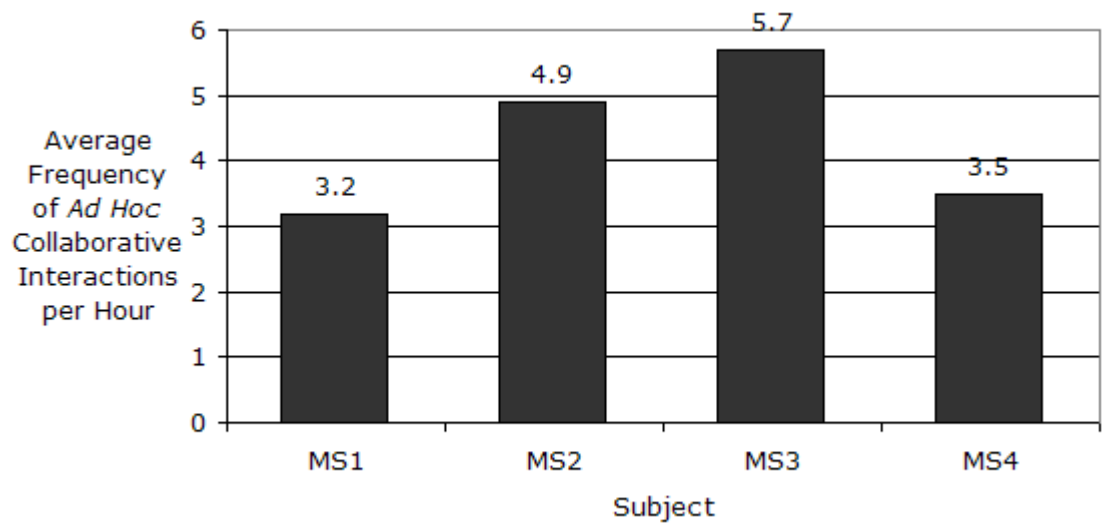


Figure 4.4 Average frequencies of *ad hoc* collaborative interactions per hour by subject

Figure 4.5 illustrates the duration distribution of *ad hoc* collaborative interactions for the whole observed sample.

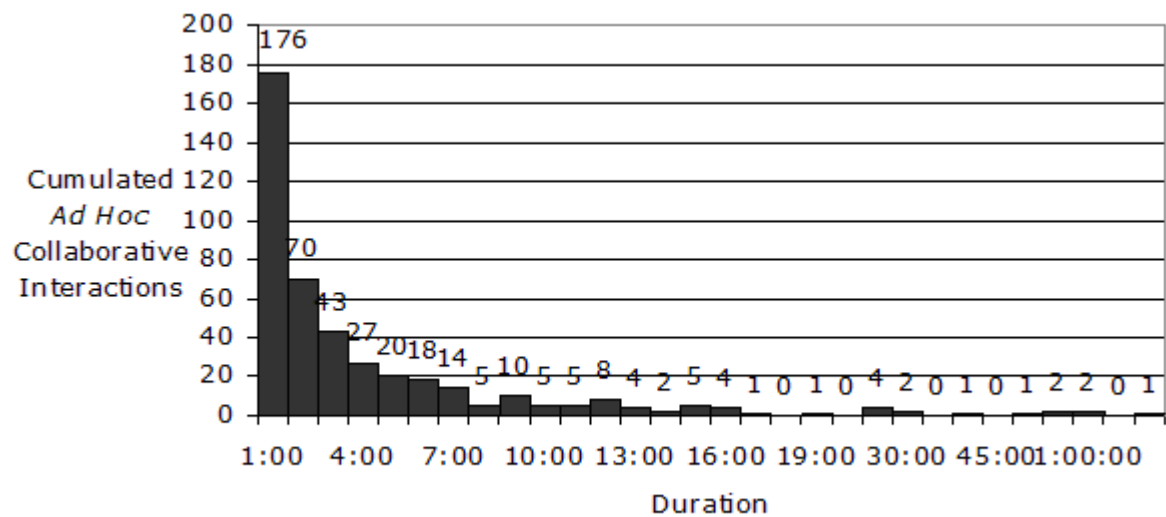


Figure 4.5 Cumulative numbers of *ad hoc* collaborative interactions as a function of their duration

Almost half the *ad hoc* interactions last less than a minute, and very few last more than 15 minutes. This tends to confirm the accepted practice of allowing spontaneous interruptions for team members to answer their team-mates' questions, knowing that they will not last very long and that the team members will shortly resume their own activities.

Figure 4.6 shows the percentage of interactions initiated by each subject. For example, MS1 was involved in 110 *ad hoc* interactions, of which he initiated 68, which amounts to 62%. The rookie (MS1) initiated almost twice as many *ad hoc* interactions as the other team members. This makes sense, given the nature of the rookie's position as a newcomer to the team and on his way up the learning curve. His efficiency depends on gathering more information than the other members. These data may quantify the adage that adding someone new to a project that is experiencing delays will just delay the project still further, since many *ad hoc*

interactions will have to be initiated by the newcomer to catch up with the team's level of expertise.

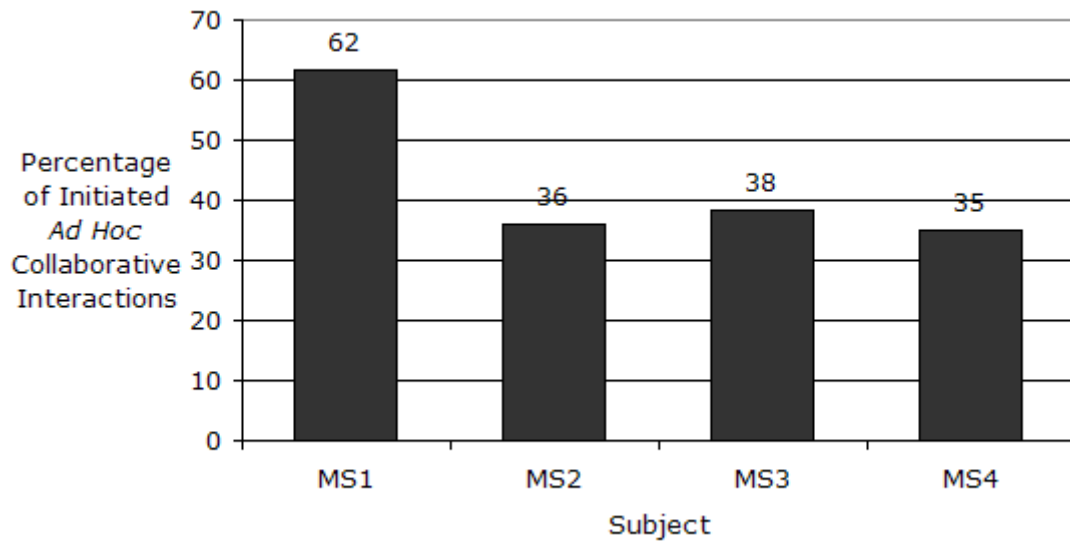


Figure 4.6 Percentage of interactions initiated by the subject

It is interesting to observe that old team-mates (MS2, MS3, MS4) reach a kind of equilibrium in terms of initiating *ad hoc* interactions. Everyone needs to interact at the same level.

Figure 4.7 shows the percentage of involvement of each subject over the total number of interactions. For example, of the total of 431 interactions observed, MS1 was involved in 110 of them, that is, almost 26%, which is less than the percentages for the other subjects. The leaders (MS2, MS3) occupy a pivotal position within the team based on the number of interactions in which they are involved.

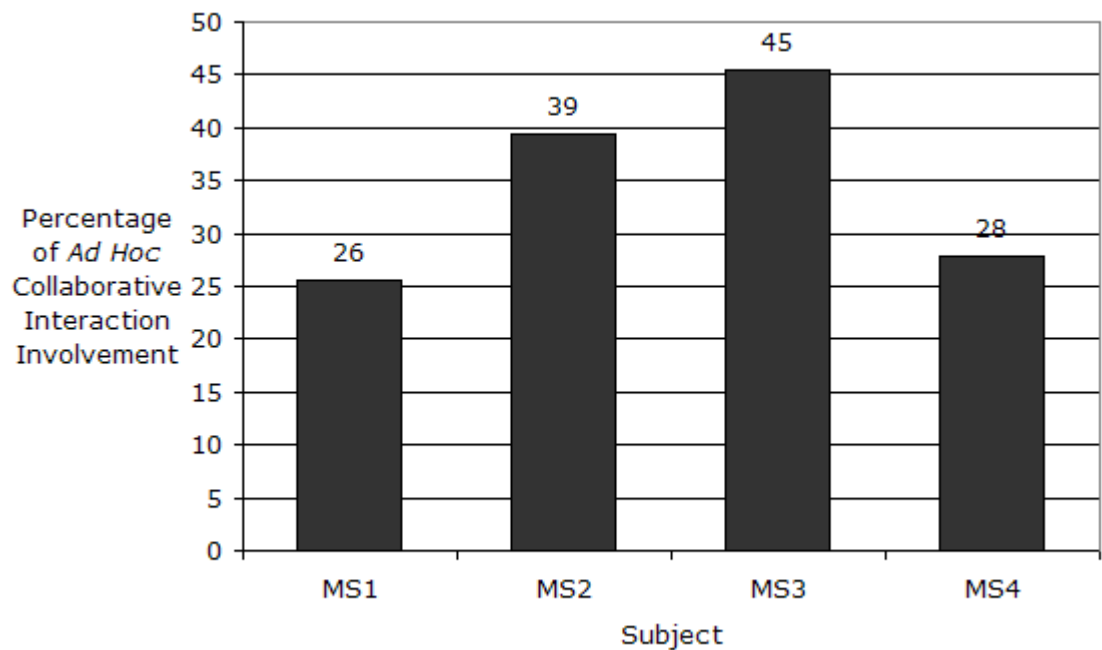


Figure 4.7 Percentage of interaction involvement per subject

Figure 4.8 combines several of the results that were presented in the network of *ad hoc* collaboration between the subjects and other stakeholders. This graph underlines the main characteristics of the four subjects and their respective roles. The size of the bubbles represents the relative proportion of all interactions in which each subject was involved, as shown previously. The size of the arrows depicts the relative number of interactions initiated by the subject from whom the arrow originates, and the relative number of interactions directed to the subject towards whom the arrow points. The darker arrows indicate the leading direction for each communication channel. For example, the communication channel MS1-MS2 has 10 communications through MS2 and 5 through MS1. Arrows also point from and towards the outside team, which is composed of the other eight people who were also part of the observed team and ten more individuals in the organisation who

interact with the observees. Each arrow pointing from or towards the outside team refers to the total number of *ad hoc* interactions with any one of these eighteen individuals. For example, MS1 initiated 31 *ad hoc* interactions with someone outside the team of the four observees and 29 *ad hoc* interactions originated from someone outside the team.

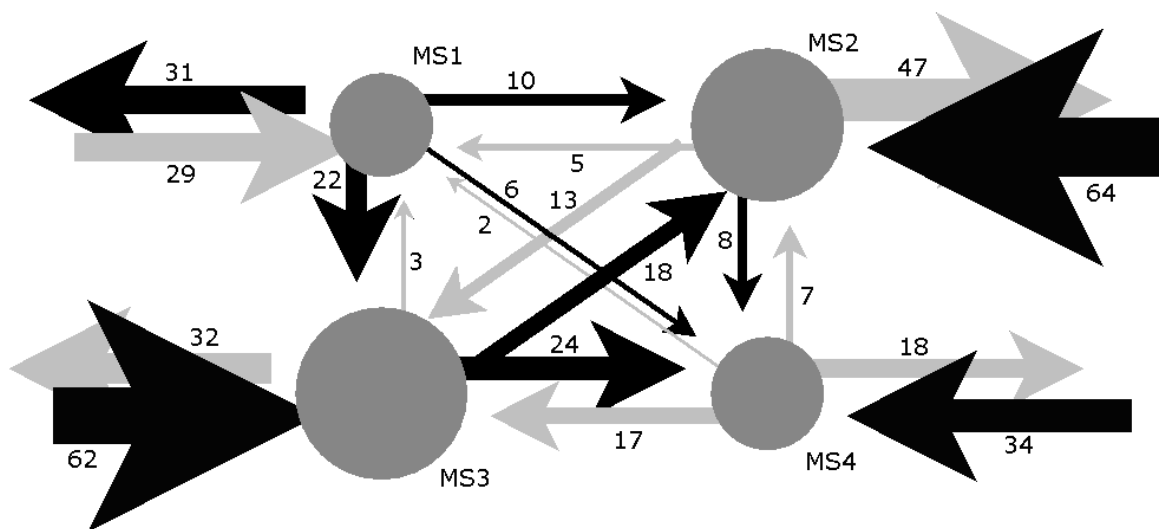


Figure 4.8 Network of *ad hoc* collaboration between the subjects and other stakeholders

Figure 4.8 presents in detail the *ad hoc* interaction patterns between the observees and people outside the team. The rookie (MS1) extracts information from the team. He mostly initiates *ad hoc* interactions to obtain information, as is shown by the fact that there are dark arrows leaving MS1 and he is rarely involved in responding to requests from within the observed team. This is a quantitative demonstration of the impact of a newcomer on the team dynamics.

The cooperator (MS4) experiences opposite situation, as all the dark arrows are pointing towards him, which indicates that people are asking him questions. He is

more often being interrupted by *ad hoc* interactions than initiating them himself. This pattern quantitatively confirms his cooperator status. He is collaborating in peer-to-peer relationships with his colleagues even though he is not involved in a significantly higher percentage of interactions than the rookie.

The reference (MS3) displays an interesting pattern: although questions are usually referred to him by someone in the outside world, he himself often refers them to MS2 or MS4. We may recall that MS1 is a junior team member and, as such, he is not likely to have answers to the questions posed by MS3. These data quantify the importance of the reference role within the team dynamics, in this case, to the extended team.

The coach (MS2) is the main *ad hoc* communicator with people outside the team. The input from outside the team is almost as that from MS3, but, in this case, he is also fairly often referred to internally by the other three participants. We may recall the formal leadership role of MS2 in his position as project manager and the informal leadership role of MS3 as a source of information and knowledge for his team-mates. Accordingly, Figure 4.8 constitutes the quantitative demonstration of the central place of leaders within a team.

The four observed subjects are members of a larger team composed of twelve people. We observed their *ad hoc* interaction behaviour as part of this larger team. Figure 4.9 shows the integrated data for the whole team dynamics. For example, MS1 responded to 39 *ad hoc* interactions from any of the team members, and he initiated 69 *ad hoc* interactions. We can compute the ratio of interactions for which

there was a response to the number of interactions initiated. A ratio greater than one could mean that the subject is providing more information to his team-mates than he requires from them. Figure 4.9 shows the ratios for the four observees. We have seen that MS2, MS3 and MS4 add information to the team's knowledge by almost the same ratio, while MS1 absorbs information from his team-mates.

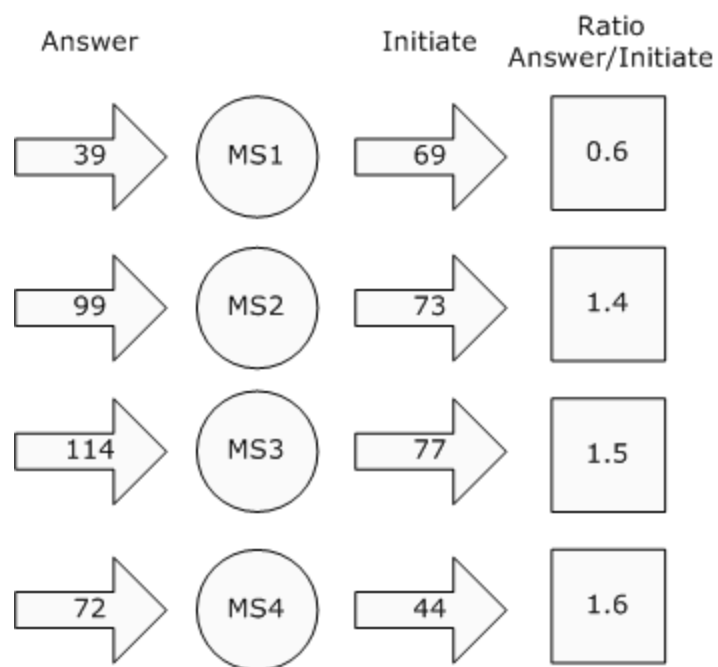


Figure 4.9 Ratio of *ad hoc* collaborations that contribute to knowledge-sharing

4.6 Concluding Remarks

This paper presented an empirical study conducted in an industrial software engineering setting with the objective of measuring and analysing *ad hoc* collaborative activities observed within a professional software development team. *Ad hoc* interactions have been observed by previous studies, but, up to now, there has been little published research on this type of activity, even though it can

account for a fairly large proportion of the time spent on a software project. Observational approaches borrowed from the social sciences have enabled us to quantitatively measure the activities sustaining *ad hoc* interactions.

Such studies are important for many reasons. First, they provide a deeper understanding of human activities within a software development team, enabling us to better understand and quantify the various roles of the participants. Although this case study is based on the observation of a specific organisation, which, at the same time constitutes the primary limitation of this type of research, the results can be extrapolated to a broader set of contexts such as any software development organisation or potentially to any team setting with similar dynamics.

Second, they provide some quantitative basis for evaluating the cost of *ad hoc* interactions with respect to the various roles. The *reference* subject plays an important role in collaborating with the team and with other people. A significant part of his time is devoted to this activity. Managers should take this into account when planning the tasks of a *reference* subject. New team members need to reach team-level expertise, and, to do so, they need to communicate on an *ad hoc* basis even though formal training may be available. *Ad hoc* interaction is akin to “just-in-time learning”. Some people are friendly and collaborate with everyone such as the *cooperator* or are informal gurus and become resources for all, such as the *reference* subject of this study, who followed this pattern to some degree. The *reference* and the *cooperator* are important in teamwork because of their knowledge or ability to help others. Undue pressure on them for task completion is likely to affect some of the team members since a number of subjects rely on them. It is

acknowledged that managers are fairly busy. In this study, we can see that the manager's *ad hoc* interactions outside the team are more important than those with the observees on his team.

One of the major results of this study is the quantitative evaluation of the *ad hoc* collaboration, which requires a great deal of time on the part of the team members and is rarely taken into account in task assignment. Moreover, there is a relationship between the role people play within a team and the time they spend in *ad hoc* collaboration activities.

Ad hoc collaboration enables people to share their knowledge or validate their understanding in a just-in-time paradigm. Face-to-face communications seem very important, as illustrated by the lack of use of emails or other electronic media available for communicating. The social aspect of face-to-face *ad hoc* collaboration has not yet been explored, but we suspect that it is fairly important and that people need human contact and body language to express themselves.

Although this study was performed in a software development environment, we feel that face-to-face behaviour may be of broader interest. We speculate, therefore, that any teamwork based on creativity and where the exchange of knowledge is important will be characterised by similar behaviour. The major characteristic of face-to-face behaviour revealed in this study is the opportunistic nature of the exchanges, which is associated with the creative aspect of the tasks to be performed. Another characteristic is link between this behaviour and the roles of the team-mates, which are not unique to the software development environment.

Newcomer, guru and manager alike are likely to behave in the same way, regardless of the project domain. These findings are probably valuable in any environment where the tasks to be performed are intellectual and rely on the sharing of knowledge. However, the physical set up, which is an open-plan office, may have a measurable impact on the exchange patterns.

This work is based on an exploratory case study, and attempts to provide a deeper understanding of the process underlying face-to-face exchanges. It focuses on finding and quantifying the parameters that characterize the exchange process. Explanatory cases will be needed to conduct causal studies, mainly to test theories based on exchange patterns. Experimental research could also be conducted based on the Goal-Question-Metrics. These two approaches can use extensive statistical analyses to validate theories or models of exchange patterns.

The next step in this research project is to analyze the content of the *ad hoc* collaboration. The sharing of information is the basis for teamwork. However, preliminary studies indicate that it is only one of the objectives *ad hoc* collaboration, others being cognitive synchronization, problem solving and coordination, for example.

This study raises many questions. *Ad hoc* collaboration is very time-consuming, since a team member's work is interrupted to listen to his team-mate's request. Would it be efficient to restrict *ad hoc* collaboration within the workplace? For example, should *ad hoc* interactions be limited before 11:00 a.m.?

Global software development scarcely takes into account the need for people to interact on an *ad hoc* basis. This study stresses the fact that, although a project can be divided up and the parts distributed among various teams, teamwork needs *ad hoc* interaction, and it is very human to engage in it.

Software development is becoming so complex that team activities are required in most projects. Team activities based on human interaction are a requirement for the emergence of new ideas and designs. The social sciences have proven sets of methods and research paradigms for studying human interactions. This paper illustrates how these approaches could be used to derive the exchange pattern of *ad hoc* interaction in a team project. Patterns and theories emerging during the course of the research and reported here will now be validated by confirmatory research using these approaches.

4.7 Acknowledgements

This research would not have been possible without the agreement of the company within which it was conducted, and without the generous participation and patience of the software development team members through whom the data was collected. To all these people, we extend our grateful thanks. Thanks also to Michel Martin for his participation in the coding validation activities. This research was supported in part by NSERC grant A-0141.

CHAPITRE 5

AD HOC COLLABORATION IN A SOFTWARE DEVELOPMENT SETTING – WHY AND HOW?

Sébastien Cherry, Pierre N. Robillard

Department of Computer and Software Engineering

École Polytechnique de Montréal, C.P. 6079. succ. Centre-Ville, Montréal, Canada

Phone (514) 340-4711 ext. 4238. Fax (514) 340-5139

{Sebastien.Cherry, Pierre-N.Robillard} @ polymtl.ca

5.1 Abstract

Team members in a co-located software development environment spend a significant amount of their time engaged in forms of ad hoc collaboration, which is defined as a spontaneous and informal exchange between two or more participants. This exploratory research is among the first attempting to characterize this type of activity, elucidate participants' motivations, and describe the communication that ensues. Our purpose is to achieve a better understanding of the nature of ad hoc activities with a view to improving collaboration processes involving software developers evolving in a co-located environment. In pursuit of this objective, the paper provides insight into suitable practices that could make collaboration among team members much more effective.

5.2 Introduction

Software developers increasingly acknowledge the impact of human communication on the software development process. The emergence of the Agile philosophies confirms that software developers prefer face-to-face communication to writing lengthy artifacts. However, few studies look into the usefulness of face to face communication, and specifically *ad hoc* collaboration, which is defined as any verbal exchange, conversation, or work conducted informally by two or more developers on an *ad hoc* basis and initiated unexpectedly. A typical example is illustrated in the following scenario, where Martin and Jean are working independently, and Martin unexpectedly addresses Jean: "Hey Jean, could you please come and check this error?" Both will then engage in *ad hoc* collaboration which will end at the conclusion of the interaction.

How time-consuming is this type of collaboration? Most studies of which we are aware have been conducted in traditional software process environments. Seaman et al. studied the communication aspect of software development (Seaman & Basili, 1997), while Grinter et al. stressed the notion of coordination (Grinter et al., 1999) and Robillard et al. focused on the concept of collaboration (d'Astous, Robillard, Detienne, & Visser, 2001), three concepts that are closely related.

In an early case study aimed at determining how developers spend their time during a software development project, Perry et al. (Perry et al., 1994-) reported that 75 minutes per day per developer are spent on informal communication. In one of our previous studies (Robillard & Robillard, 2000), we found that almost 40% of the

time spent by developers is devoted to *ad hoc* collaborative activities. *Ad hoc* communication, which is the subject of this article, is the spontaneous communicative activity of collaboration among co-workers.

Some authors consider *ad hoc* communication to be an intrusion, which Jett and George (Jett & George, 2003) define as “an unexpected encounter initiated by another person that interrupts the flow and continuity of an individual’s work and brings that work to a temporary halt.” Intrusions tend to be immediate in nature, and much of their disruptive impact derives from the perceived need to respond promptly to the needs of the interrupter.

Other research has documented the fragmented nature of software development work in terms of frequent interruptions and coordination (Chong & Siino, 2006). Ko et al. (Ko et al., 2007) analyzed detailed activity logs for the information that developers sought. They consider *ad hoc* face-to-face activities and e-mails as causes of task-switching. Seaman et al., and more recently Gutwin et al. (Seaman & Basili, 1998; Gutwin, Penner, & Schneider, 2004), have demonstrated that co-location, participant mobility, and spontaneous communication are major factors in determining the quality of awareness information and in facilitating the solution of coordination problems.

The case study presented in this article was conducted in an industrial software engineering setting, and relies mostly on participant observation. The goal was to observe a co-located team where developers are free to interact with one another.

The methodological approach used in this study is presented in the next section, followed by the results ensuing from it. The approach reveals the characteristics of *ad hoc* communications, and, more specifically, their motivations and the communication channels used. The impacts of findings on the software development process and environment are discussed.

5.3 Methodology

Our data were taken from continuous audio-video recordings of activities of the members of a software development team in a large software development organization. This team was composed of 12 individuals (1 project manager and 11 software developers) ranging widely in age, with varying levels of schooling (from a Bachelor's degree to a Ph.D. in computer sciences and engineering), and individual experience ranging from 2 to 16 years in the field and from 9 months to 5 years of service in the company. The team is continually evolving in the organization, which has a highly standardized software development process comprising several thousand developers in a number of countries around the globe. In spite of the size of the organization, the setting has the attributes of smaller organizations, as development is shared among several small teams of up to 15 members each, often located at a single site. Their need to cooperate stems from the fact that they are grouped by functional expertise, as well as by product area or software component, both of which are tightly integrated to form an enormous solution of millions lines of code from a number of different technologies.

The observations are based on half-day work sessions, in the form of nearly 200 hours of audio-video recordings and more than 2,500 e-mail messages exchanged between team members.

The e-mails were gathered with the help of triggers defined in the messaging software used by the company. Both sent and received e-mails were collected for cross validation.

The audio-video recording of half-day sessions provided data on *ad hoc* collaborative activities in the form of face-to-face (FtF) communications and telephone conversations. The capture of e-mails enables analysis of electronic mail exchanges. These three collaboration channels were identified as the sole modes of communication used by software developers on an *ad hoc* basis in this organization. Of course, formal and called meetings are also used for collaborative activities, but are not considered in this study, which targets only *ad hoc* collaboration.

For this study, we selected 12 representative half-day sessions of the 23 recorded. These selected sessions are evenly distributed over the two months of the recording time and account for 35 hours of audio-video recording, resulting in 404 FtF communications, 110 e-mail exchanges, and 32 telephone conversations. The full recording analysis was made on a subgroup of four participants from the 12 co-located team members. Further details on the methodology used can be found in previous publications by the same authors, which analyzes the patterns of *ad hoc* communication among the participants (Cherry & Robillard, 2008).

5.3.1 Data Modeling

Figure 5.1 shows the characteristics of *ad hoc* communications. They are normally held by teammates (who) with the objective of working together on particular matters or topics (what) in the form of particular cognitive activities (why) by means of various communication channels (how).

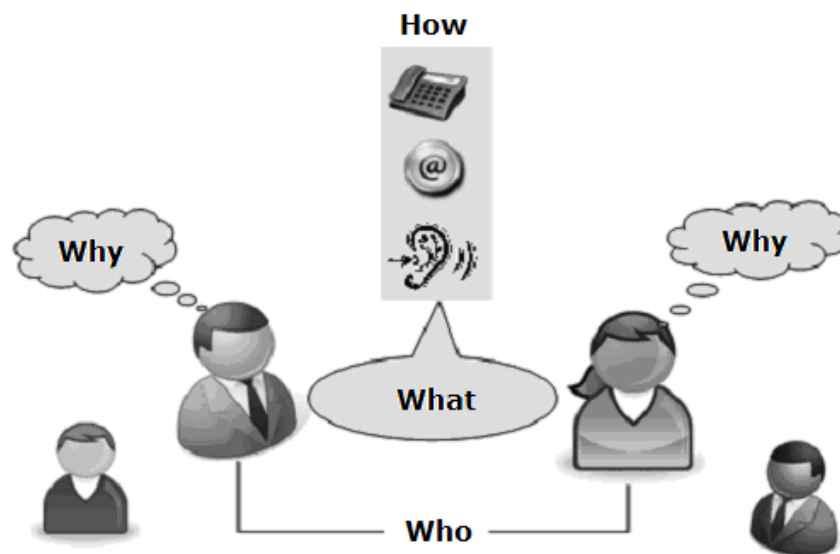


Figure 5.1 The characteristics of *ad hoc* communications

Figure 5.2 presents a UML representation of the model used for encoding the audio recording. This model was inspired from a combination of the linguistic and object-oriented paradigms (d'Astous & Robillard, 2002a), and defines the observational units and relevant attributes.

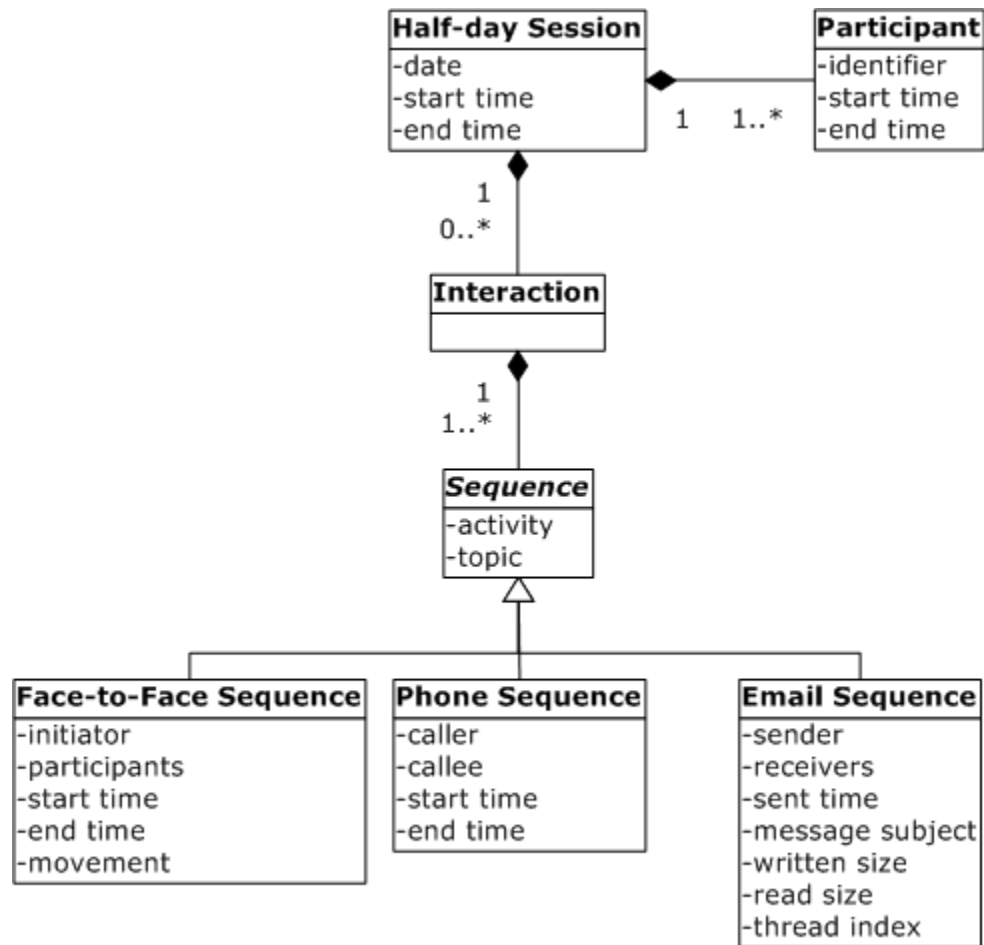


Figure 5.2 Data model designed for encoding the raw data

A half-day session begins either in the morning or in the afternoon after the socialization period, and lasts 2 to 3 consecutive hours. An *interaction* is a communicative unit of one or more sequences that presents evident internal continuity, while being distinct from what precedes it and what follows it (Kerbrat-Orecchioni, 1998). These interactions are derived from the underlying sequences. The *sequence* entity constitutes the smallest unit analyzed in an interaction. It takes the form of a block of one or more exchanges between participants that show a high degree of semantic or pragmatic coherence, i.e. they stand in for the performance

of a particular type of activity (why) on a particular topic (what) and are composed of different attributes, such as the participants (who), based on the particular communication channels used (how).

5.4 Channels of *Ad Hoc* Collaboration

While a previous article (Cherry & Robillard, 2008) looked at *ad hoc* collaboration from a social point of view, in which participants occupy different roles (who) based on their formal and informal leadership and influence within the team, this work goes further by describing the communication channels used (how) during *ad hoc* collaborations realized by the whole team, as well as by the observed participants individually.

Analysis of the time spent on *ad hoc* communication by the four observed participants shows that more than a quarter (28%) of their time was spent on informal spontaneous *ad hoc* FtF communication. Telephone *ad hoc* communication occupied only 1% of their time, indicating that this channel is not significant enough in this setting to warrant more attention. The rest of the time (71%) was spent on other types of activities, such as individual activities, formal meetings, etc. The above results do not take into account e-mail exchanges, since it was not possible to measure the time spent on the tasks of reading and writing e-mails.

These data confirm the importance of the phenomenon found in previous studies (Perry et al., 1994-; Robillard & Robillard, 2000), which in this study accounts for almost 2 hours and 20 minutes per 8-hour workday per participant.

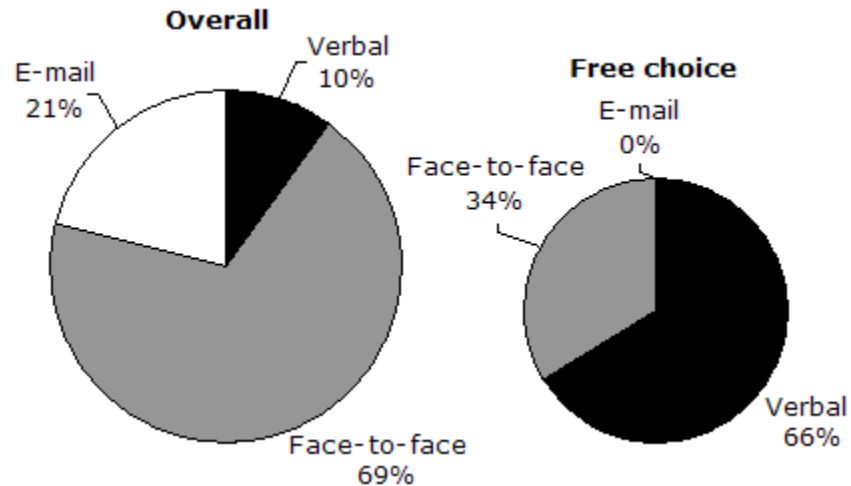


Figure 5.3 Frequency distribution from the Face-to-Face, Verbal, and E-mail *ad hoc* collaboration channels

Figure 5.3 illustrates two related results on channel usage frequency for *ad hoc* communication. The pie chart on the left shows the channel usage frequency distribution for the e-mail and FtF communication involving at least one of the four observed participants. FtF interactions, which account for 28% of the overall observation time, require 69% of all channel usage. Verbal *ad hoc* communications, which occur when participants are verbally communicating while seated at their desks, account for 10% of channel usage. E-mails occupy 21% of channel usage frequency.

The pie chart on the right presents the relative frequency of channel usage when the participants are free to choose among the three channels, i.e. when the initiator had complete flexibility to engage in the interaction via one channel or another without any constraint, such as the distance between himself and his interlocutor,

the time needed to cover the interaction, etc. The pie chart on the right represents 14% of the 514 interactions represented in the pie chart on the left.

We observed first that e-mail usage drops to 0%, i.e. no one communicated naturally by e-mail, when an FtF or a verbal interaction was possible, and second that verbal interactions are clearly dominant, with a relative frequency of 66%. These results quantify the strong participant preference for verbal interaction.

FtF interaction is preferred when the interactions require more time and/or visual feedback from the interlocutor. It worth noting that e-mails are almost exclusively used (97%) when there is no other alternative, i.e. to communicate with a distant stakeholder, to send a file, to forward a previous e-mail to somebody else, or to contact several stakeholders at the same time.

5.4.1 Role Definition

Case study participants were described in terms of the role or position that each of them occupies within the team, and this is the key to understanding the results subsequently presented. These roles are often informal and implicit, and not formally acknowledged by the main players themselves. Based on qualitative observations, the roles of the four participants identified are the following (Cherry & Robillard, 2008):

- MS1: the software developer recruit on the team.
- MS2: the project manager who occupies the formal leadership position.

- MS3: the individual responsible for configuration management of the software built by the team, his informal leadership being rooted mostly in his knowledge and expertise.
- MS4: an individual with no specific role in the team, formal or informal, but who can be seen as the embodiment of an average developer.

Table 5.1 presents *ad hoc* communication channel and sub channel usage in terms of frequency for each participant. Note that the *total* column indicates the total occurrences of interactions of the specific channel type, no matter who the participants were. For example, there is a total of 49 verbal observations, which require at least two participants each. There are also some verbal interactions involving participants who are not among the four being observed. The numbers in bold indicate the highest number of contributing occurrences for a given channel. Two sub types of FtF interaction involve two participants (F2F) and many participants (FnF). F2F and FnF interactions add up to FtF interactions. The electronic communication channel is divided into three sub types, which are E-mail (person to person), CC (copy to), and FWD (forward to).

Table 5.1 *Ad hoc* collaboration channel occurrences per participant

	MS1	MS2	MS3	MS4	Total
Verbal	2	18	23	30	49
Face-to-face	105	138	164	89	355
F2F	94	102	137	63	299
FnF	11	36	27	26	56
E-mail	25	86	32	25	110
CC	1	20	8	1	38
FWD	0	14	4	2	22

The first contrasting results are the divergent involvement of MS4, who was involved in the most verbal interactions (30) and the fewest FtF interactions (89). This could be explained by the fact that MS4 is located centrally within the team's workspace. MS2 and MS3, who are located next to MS4, are not only involved in a significant number of verbal interactions, but also a high number of FtF interactions, with 138 and 164 occurrences respectively. We recall that MS2 (formal) and MS3 (informal) both assume some form of leadership.

A second interesting result is the relatively large number of person-to-person, CC, or FWD e-mails (86) in which MS2 was involved, compared to his colleagues. As project manager, MS2 needs to network with several distant stakeholders, and so requires the e-mail interaction channel. This also explains his frequent involvement in CC or FWD e-mail communications with his subordinates.

5.5 Types of *Ad Hoc* Collaborations

Results regarding the types (why) of *ad hoc* communication provide a deeper understanding of the motivations of the phenomenon by further qualifying the relationships between the participants, their roles, and the communication channels they use.

We define five types of *ad hoc* communication:

- Information-sharing occurs when a participant shares information without any prior request from someone else. For instance: "By the way Michel, I modified your function ABC in class EFG in order to cover this case."
- Cognitive synchronization occurs when participants exchange their understanding on a topic to validate their mental representation of the subject being discussed. This exchange of knowledge is always initiated by a request. Example: "Hey Chris, could you please tell me how we should configure this?"
- Development occurs when participants explain their work on a particular task of the project. Example: "I would implement this feature by using the component block xyz. Then, [...]."
- Coordination occurs when participants want to schedule meetings or activities, define priorities, or give a status report on an ongoing task. Example: MS1 asks, "When would you be available to meet to discuss that?" MS3 replies, "After lunch." MS1 responds, "That would be fine."

- No relevant interaction occurs when participants are talking about a topic that is not related to the project. Example: "Hi Joe! How was your weekend?"

Table 5.2 details the relative frequency of the *ad hoc* collaboration types identified (why) for each communication channel (how).

Table 5.2. Frequency distribution of *ad hoc* collaboration types per channel

	Information Sharing	Cognitive Synchro.	Develop.	Coordination	Not relevant
Verbal	12%	43%	5%	17%	23%
Face-to-face	7%	49%	23%	8%	13%
F2F	8%	45%	26%	8%	13%
FnF	2%	69%	10%	5%	14%
E-mail	22%	26%	20%	27%	5%
CC	13%	29%	32%	26%	0%
FWD	64%	0%	4%	27%	5%
Total	11%	42%	21%	14%	12%

A salient result from Table 5.2 is the dominance of cognitive synchronization for any communication channel. Verbal interaction is mostly used for cognitive synchronization (43%) and coordination (17%). FtF is mostly used for cognitive synchronization (49%) and development (23%). FnF is essentially used for cognitive synchronization (69%), while F2F involves some development activities (26%). It is interesting to observe that information-sharing is of interest only in the case of verbal communication.

E-mails are used irrespective of the *ad hoc* collaboration type. FWD e-mails are used mostly for information-sharing (64%), as could be expected.

Table 5.3 presents types of *ad hoc* collaboration (why) with observed participants (who). The upper part of the grid shows the distribution of *ad hoc* collaboration types from a participant perspective (the percentages on the lines add up to 100%), while the lower part shows the distribution from the perspective of collaboration type (percentages in the columns add up to 100%).

Table 5.3 Frequency distribution of vocal *ad hoc* collaboration type per participant

	Information Sharing	Cognitive Synchro.	Develop.	Coordination	Not relevant	
MS1	11%	43%	15%	15%	16%	100%
MS2	17%	38%	17%	23%	5%	100%
MS3	10%	49%	24%	7%	10%	100%
MS4	10%	48%	21%	5%	16%	100%

MS1	14%	16%	13%	20%	24%
MS2	44%	27%	28%	56%	16%
MS3	25%	34%	36%	16%	28%
MS4	17%	23%	23%	8%	32%
	100%	100%	100%	100%	100%

Almost half the *ad hoc* collaboration activities for any participant involve cognitive synchronization (38%-49%). The relative weight of project leader (MS2) cognitive synchronization is less than that of the others, owing to his greater involvement in coordination (23%).

Development activities are the next reason for *ad hoc* collaboration for the experienced participants (MS3 and MS4). This type of activity takes into account almost 25% of the *ad hoc* interactions of these participants.

The lower part of the grid in Table 5.3 presents the *ad hoc* interactions from the perspective of collaboration type. The first observation is that the two roles with some form of leadership, project leader and guru, dominate the vocal collaboration type. MS2 leads in information-sharing and coordination collaboration, and MS3, who is the guru, leads in cognitive synchronization and development work.

We recall that MS2 is the leader of a 12-person team, and this observation was made on a sub team of 4 people. The leader is pivotal in the sharing of information on an *ad hoc* basis among team members. The guru, MS3, is highly involved in information-sharing too (25%). He is also a key resource in cognitive synchronization (34%), as well as in development work (36%). Finally, irrelevant interactions are mostly performed by MS4 (32%), who is centrally located, as outlined for verbal interaction. MS1, the recruit, does not dominate in any type of *ad hoc* collaboration.

These results strongly support the suggestion of LaToza et al. (LaToza, Venolia, & DeLine, 2006), to the effect that communication patterns are sometimes polarized towards individuals who are likely to possess legacy knowledge about the ongoing project. Such gurus may act as an attractive pole for *ad hoc* interaction on the part of team members digging for specific information.

5.6 Discussion and Concluding Remarks

5.6.1 Major findings

Cognitive synchronization appears to be the main motivation (why) for initiating *ad hoc* collaboration. Participants need to synchronize their mental model of the activities they are performing with that of their colleagues. Since it is *ad hoc*, this type of interaction could be perceived as *just-in-time* knowledge-sharing or validation. Cognitive synchronization accounts for up to almost half the entire effort spent on any *ad hoc* interaction, and, based on the average time spent on it, this activity accounted for a total of 1 hour and 10 minutes per working day. Because of the *ad hoc* nature of this cognitive synchronization, it could be difficult to replace it with some sort of called or scheduled meeting.

Verbal interaction is the preferred communication channel (how), FtF is next in importance, and e-mails are used only when no other alternatives are available.

A previous study based on interaction patterns indicates that knowledge is not evenly distributed among team members. Some participants require the collaboration of their colleagues to acquire knowledge, while others act as a source of knowledge. For example, we observed that the recruit (MS1) initiated 62% of his *ad hoc* interactions, while the other participants initiated only 36% of theirs (Cherry & Robillard, 2008).

The type of leadership engaged in by the participants, as well as their knowledge, and the teammates' preference for verbal interaction suggest a physical setup

where these participants are centrally located in an open workspace. For instance, placing a new recruit right next to the guru could facilitate knowledge-sharing, which is essential to help the newcomer become autonomous quickly. It could be counterproductive for the person with some form of leadership role to have an isolated corner office, for example, or for the new recruit to have a workspace that is distant from those of his teammates. In light of these results, a software development team's workspaces should be designed to facilitate collaboration, with particular emphasis on ease of communication (how) among teammates, either verbally or through face-to-face interaction.

These observations also suggest that adding a new recruit to a team without proper training on the specific project could be time-consuming and a drain on team resources. The new recruit is essentially initiating *ad hoc* interactions for the purpose of synchronizing his understanding of the project with that of his teammates, and will rely on the guru's time to obtain details on the project.

In most organizations, project leaders who are part of the development team have task assignments that take into consideration their management activities. This study shows that similar considerations should be given to team members acting as gurus or references for the project who spend time in *ad hoc* interactions initiated by teammates for the benefit of the team's performance on the project.

Ad hoc interaction is an essential tool for team-driven activities like software development. However, it involves at least two people, the initiator and the respondent. The impact of an interruption on the respondent's activities has not

been measured. The social nature of *ad hoc* interaction is also an important factor. After some time has been spent on individual tasks, most people need to interact with someone. It is reasonable to suppose that some of the *ad hoc* interactions, which may or may not be relevant, are probably derived from these needs.

Work is in progress to analyze the content of *ad hoc* interactions. We hope that this will provide some indication of the importance of that content for the organization, and of how the relevant information could be traced or recorded.

5.6.2 Validity Concerns

The validity of the observed results relies on the fact that they faithfully and reliably represent reality. This study was not an experiment where the various parameters could be controlled, and this paper reports observations performed in a real professional environment in which participants are carrying on their day-to-day activities. The salient outcome of these observations, regardless of the specific setup of the organization, is a better understanding of the *raison d'être* of *ad hoc* collaboration and how it occurs.

The definitions and concepts used in the case study presented in this paper have been defined to be applicable to other environments of software development. Moreover, although the detailed results may vary from context to context, we feel there are grounds to believe that the patterns uncovered and derived from the results presented are likely to be generalizable to many software development settings.

5.7 Acknowledgments

This research would not have been possible without the agreement of the company in which it was conducted, and without the generous participation and patience of the software development team members from whom the data were collected. To all these people, we extend our grateful thanks.

CHAPITRE 6

EMPIRICAL ANALYSIS OF *AD HOC* COLLABORATION PATTERNS AND CONTENT IN A CO-LOCATED SOFTWARE DEVELOPMENT ENVIRONMENT – A CASE STUDY

Department of Computer and Software Engineering
École Polytechnique de Montréal, C.P. 6079. succ. Centre-Ville, Montréal, Canada
Phone (514) 340-4711 ext. 4238. Fax (514) 340-5139
{Sebastien.Cherry, Pierre-N.Robillard} @ polymtl.ca

6.1 Abstract

It is now acknowledged in the literature that ad hoc collaboration between teammates, defined as spontaneous and informal exchanges between two or more participants, constitutes a sine qua non in co-located software development settings. However, there is no consensus among researchers and practitioners on the way to deal with this time-consuming element in software projects. In our opinion, this is because the intrinsic raison d'être of ad hoc collaboration has not yet been uncovered. The research presented in this paper is aimed at moving this debate forward. It consists of an empirical study in an industrial software engineering setting that measures ad hoc collaboration observed between developers in their working environment. This exploratory research is among the

first to attempt to characterize this type of collaboration, discern the patterns, and describe the content of the communication that emerges. The purpose is to better understand the nature of ad hoc collaboration to ultimately improve the collaborative processes in a co-located software development environment, while providing good avenues for improvement in a de-located context as well.

6.2 Introduction

Collaborative work performed in software engineering is generating interest nowadays, as researchers increasingly acknowledge the impact of human communications on the software development process. For instance, Seaman and Basili studied the communication aspect of software development (Seaman & Basili, 1997), Herbsleb, Grinter, and Perry stressed the notion of coordination (Grinter et al., 1999),(Herbsleb & Grinter, 1999) and Robillard and Robillard focused on the underlying concept of collaboration (Robillard & Robillard, 2000).

The concepts of communication, coordination, and collaboration are interrelated, since coordination is a specific type of collaboration, and communication actually constitutes a concrete manifestation of collaboration. Besides these conceptual views, software engineering processes formalize several forms of collaboration between software developers: peer-review meetings, walkthroughs, inspections, whiteboard designs, and so on.

Numerous studies have revealed that a significant amount of time is spent by developers on informal and *ad hoc* collaboration. Perry, Staudenmayer, and Votta

(Perry et al., 1994-), for instance, reported in their case study aimed at determining how developers spend their time during a software development project, that 75 minutes per day per developer is spent on informal communication. In another case study, Robillard and Robillard (Robillard & Robillard, 2000) found that 41% of the time spent by developers is devoted to forms of *ad hoc* collaboration, defined as informal meetings of team members to work on a given task or simply to share information on an *ad hoc* basis. By informal and *ad hoc* collaboration or communication, we mean any talk, conversation, or work done informally by two or more developers on an *ad hoc* basis and initiated unexpectedly.

Some authors consider *ad hoc* collaboration as intrusive. Jett and George (Jett & George, 2003) define an intrusion to be “an unexpected encounter initiated by another person that interrupts the flow and continuity of an individual’s work and brings that work to a temporary halt.” Intrusions tend to be immediate in nature, and much of their disruptive impact derives from the perceived need to respond promptly to the needs of the interrupter. Other research has documented the fragmented nature of software development work, with frequent interruptions and coordination (Chong & Siino, 2006). Ko et al. (Ko et al., 2007) analyzed detailed activity logs for the information that developers sought. They consider *ad hoc* face-to-face activities and e-mails as causes of task-switching. DeMarco and Lister (DeMarco & Lister, 1999), for their part, maintain that interruptions break what they call the *mental flow* in developers’ minds, and that time is required for them to return to the productive state they were in before the interruption. For that reason, they suggest limiting interruptions.

In contrast, Seaman et al., and more recently Gutwin et al. (Seaman & Basili, 1998) (Gutwin et al., 2004), have demonstrated that the mobility and spontaneous communication of co-located participants are major factors in determining the quality of awareness information and facilitating the solution of coordination problems.

In order to achieve a better understanding of *ad hoc* collaboration within a software development team, we believe it is imperative to go further than previous studies and analyze the inherent patterns and content of collaborative interactions. Such an understanding will provide the keys to improving the design of the software development process and the environment for co-located software teams. These improvements will, in turn, help meet the needs of collaboration in an actual software engineering setting and address the opposing views found in the literature and reported above. Moreover, while other studies have shown that distance raises barriers to informal communication in a de-located context, resulting in a number of coordination problems (Herbsleb & Grinter, 1999), we believe it is reasonable to expect that a better understanding of these informal communications will pave the way to further improvement in the so-called of Global Software Development area (Herbsleb & Moitra, 2001). This is what we aim to achieve with the research presented in this paper.

The methodological approach we use in this empirical study is presented in the next section, followed by the results ensuing from it, which reveal the characteristics of *ad hoc* collaboration and, more specifically, the patterns and content comprising this type of activity. The impacts of findings on the software development process and

the co-located environment are discussed in the context of co-located team activities, but we also stress the implications for de-located settings.

6.3 Methodology

The research presented in this paper was conducted in the form of a case study in an industrial software engineering setting, and relies mostly on participant observation, as described by Jorgensen (Jorgensen, 1989) and Babbie (Babbie, 2001). The goal of this case study is to observe a co-located team, where developers are free to interact with one another. Our purpose is to observe and measure *ad hoc* collaboration occurring in a real professional environment. Therefore, a co-located team of 12 developers working for an international software vendor company employing several thousand developers was investigated by an observer who acted as a real participant on the team, i.e. as a fully-fledged member of that team. The observations were made in a maintenance context, which is the situation that predominates in this large company, where the software that has been developed needs to be maintained and enhanced. The total observation period lasted several months, during which a large quantity of information was collected:

- nearly 200 hours of audio-video recordings of working sessions;
- about 2,500 e-mail messages exchanged between team members.

Exploratory research indicated that the specific attributes of the phenomenon under study may not have been completely defined. Therefore, by collecting a larger set of data, we were free to decide which specific aspects justified further exploration. The

audio-video recording approach has been approved by an independent ethics committee of both the participating organization and the École Polytechnique de Montréal. This approach was fairly unobtrusive, since the participants admitted to having forgotten the presence of the camera after only a few days. Moreover, because the recordings were started four weeks before the designated observation period, the participants had time to get used to the camera that was to become part of their work environment. The footage obtained during these “warm-up” weeks, about which the participants were unaware, was discarded so as to eliminate any bias attributable to the presence of the camera.

The electronic mail messages (e-mails) collected were gathered with the help of triggers defined in the messaging software used by the company. We collected both sent and received e-mails for cross validation, since what was sent by the sender had to have been received by the receiver in order to be considered valid.

By recording working sessions, we were able to observe *ad hoc* collaborations by means of *vocal* as well as *telephonic* communication channels. The capture of e-mails enabled us to analyze exchanges performed *electronically*. These three communication channels were identified during a preliminary ethnographic period as the only ones used by software developers to collaborate on an *ad hoc* basis in this organization.

The data analyses were based on the Exploratory Sequential Data Analysis (ESDA) technique described by Fisher and Sanderson (Sanderson & Fisher, 1994; Fisher & Sanderson, 1993-). This technique is well suited to analysis requiring the sequential

integrity of the data, as well as introducing operations for analyzing and transforming the data to obtain derived data on which the conclusions will ultimately be based. Some of these operations allow the transformation of qualitative data into quantitative data through an encoding process, which limits their variability and facilitates further operations.

Within the framework of this study, we selected 12 representative sessions of the 23 recorded. These sessions are distributed over the two months of the recording time and account for 35 hours of audio-video recording, resulting in 404 vocal, 110 electronic, and 32 telephonic communications. The full recording analysis was made on a sub-group of 4 individual participants of the 12 co-located team members. The sampling is required to keep the data analysis to a reasonable level, since this analysis of the activities performed during each recorded hour required a tremendous amount of time to complete. Consequently, following the social tradition approach described by Sanderson and Fisher (Sanderson & Fisher, 1994), a judged sampling, based on the representativeness of regular sessions, was performed on the mass of information that was collected in the field.

Preliminary studies outlining the *ad hoc* collaboration network resulting from the vocal communications has been presented elsewhere (Cherry & Robillard, 2008). Why such a network exists and how it is structured has been presented previously (Cherry & Robillard, To be published).

These previous analyses are based on the same data modelization model, which, has been completed in the next section.

6.3.1 Data Modelization

Ad hoc collaboration normally occurs when the participants (who) “collaborate” on specific matters or contents (what) at a particular time (when) in types of particular cognitive activities (why) by means of different communication channels (how). These are the elements that compose *ad hoc* collaboration, and consequently constitute the focal attributes studied through this research. We apply protocol analysis, an empirical approach detailed in previous studies on peer review and participant roles (d'Astous et al., 2001), (d'Astous & Robillard, 2002a; d'Astous, Detienne, Visser, & Robillard, 2004-) to the audio-video recordings we created in order to describe the instances of *ad hoc* collaboration observed in this study.

In preparation for the encoding phase of the analysis of the raw data collected, a model inspired by combining the linguistic and object-oriented paradigms, as used by d'Astous and Robillard (d'Astous & Robillard, 2002b), which defines observational units and the relevant attributes that should be observed in the raw data (d'Astous & Robillard, 2000), has been designed. Figure 6.1 presents the UML representation of this model.

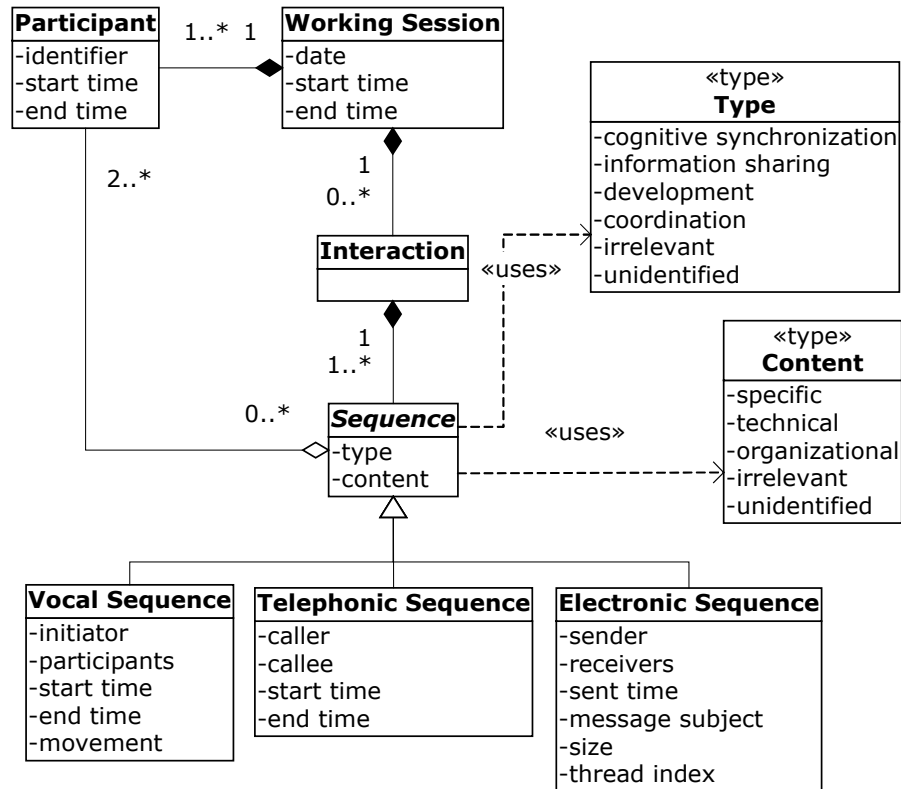


Figure 6.1 Data model designed for encoding the raw data

In Figure 6.1, a *working session* represents a particular time frame during which observations on *ad hoc* collaborative activities are made and which involves one or more *participants* with a corresponding identifier and working start time and end time. A working session is a half-day period (AM or PM) lasting from 2 to 3 continuous hours. *Interaction* is the largest unit of observation in this study. It is, as described by Kerbrat-Orecchioni (Kerbrat-Orecchioni, 1998), the largest communicative unit of one or more sequences that presents a clear internal continuity, while being distinct from what precedes it and what follows it. These interactions do not have their own attributes, since they can be derived from underlying sequences. The *sequence* entity is the smallest analyzed unit within an interaction, and takes the form of a block of one or more exchanges between at

least two participants which show a high degree of semantic or pragmatic coherence, i.e. they are related to the same topic or oriented towards the same activity. Sequences stand in for performing a particular *type* of activity (why) on a particular topic or particular *content* (what). These two attributes of sequences are shown as enumeration type in the model with predefined defined category values that have been identified in a preliminary ethnographic phase prior to the data collection and encoding phases. These values will be explained later.

Communication channels (how) have various attributes. A vocal sequence is characterized by an initiator, a start time, and an end time (when); it involves at least one other participant (who); and may involve a movement so that the sequence could take place at a particular cubicle. A telephonic sequence is characterized by a caller and a callee (who), as well as by a start time and an end time. An electronic sequence differs in that it comprises a sender and one or more receivers (who), a send time, a message subject, a size in number of words, and, finally, an index if the message was part of a thread of several messages.

6.3.2 Validity Concerns

In addition to the ESDA methodology and because audio-video recording allowed us to iterate many times on the raw data, intra- and inter-coder agreement phases of the study were conducted. This made it possible to evaluate the existence of subjectivity in the data and increase confidence in the validity of the results. A coding scheme has been defined for the five contents identified (what) and types of *ad hoc* communication (why), and two inter- and intra-coder agreement phases

have been applied to validate the data on these particular attributes. This was done because these attributes were acknowledged to constitute a greater subjective component than the other attributes. Sequences from the three communication channels formed part of the intra- and inter-coder agreement phases. While the same segmentation of the audio-video recordings was re-encoded for the vocal and telephonic sequences, the same text segments were also re-encoded for the e-mail sequences, the purpose being to measure the agreement in the categorization of the same pieces of raw data for the two particular attributes mentioned above, i.e. labeling each sequence with one of the predefined values listed in the enumeration shown in the model in Figure 6.1. The first phase involved an intra-coder agreement, where a number of encoded data sequences were re-encoded a month later by the same coder. The second phase involved an inter-coder agreement, where the same operation was performed, but this time by another coder who was able to understand the context and the jargon employed by both the participants and the primary coder. An index proposed by Perreault and Leigh (Perreault & Leigh Laurence E., 1989) was used to measure the subjectivity of the coding.

Table 6.1 Intra- and inter-coder agreement results

	Simple Proportion	Perreault and Leigh's Index
Intra-coder Agreement	0.98	0.99
Inter-coder Agreement	0.81	0.89

As shown in Table 6.1, a clear consensus emerges from the intra-coder agreement, since only 5 disagreements were noted on 227 encoded data sequences. The indices obtained through the inter-coder agreement show an agreement between the two coders with a value of 0.89, which is above the 0.7 limit, enabling us to deduce a strong agreement.

Encoders could use an *unidentified* category if they were unable to encode a particular sequence. This loophole category is necessary because of the uncontrolled setting in an industrial context. For example, uncodable interactions occurred when participants took part in *ad hoc* collaboration in the form of vocal communications outside the range of the microphones, but within the scope of the camera. Another reason is the high level of background noise at certain times, with up to 100 co-workers sharing the floor space. This *unidentified* category, which accounts for 34% of the data, reduced the data on which the results specific to content (what) and types of *ad hoc* collaborative activities (why) are based. However, based on the social tradition approach, explained by Sanderson and Fisher (Sanderson & Fisher, 1994), the validity of the results as representative of the data is preferred over statistical sampling.

6.4 Channels of *Ad hoc* Collaboration

A previous article by the same author (Cherry & Robillard, 2008) presented a social point of view (who) of *ad hoc* collaboration, in which participants take on different roles based on their formal and informal leadership and on their influence on the

team. This paper addresses more than the social roles by exploring the content of *ad hoc* collaborations, i.e. the type of activities they involve (why) and the content (what) of the discussions between the participants (who). First, we present below the communication channels (how) in which these interactions occurred.

6.4.1 Usage of Communication Channels

Figure 6.2 presents an initial outline of communication channel usage during *ad hoc* collaboration by representing the percentage of time spent by each channel. This figure shows that participants spent 28% of their time in *ad hoc* collaboration taking place on the vocal communication channel. Telephonic *ad hoc* collaboration occupied only 1% of the time. The telephonic channel will not be studied further in this analysis.

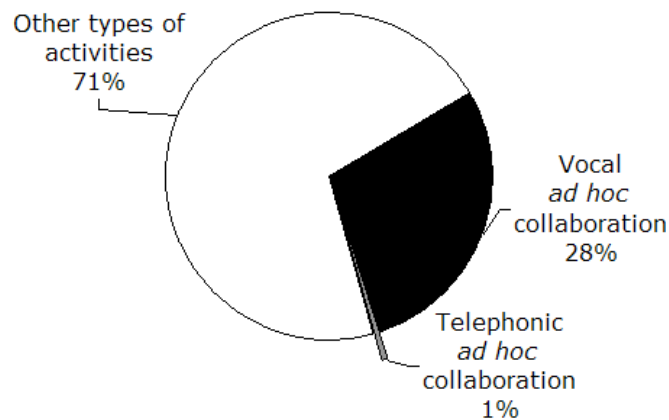


Figure 6.2 Percentage of time spent on *ad hoc* collaboration per communication channel

The 404 interactions occurring through the vocal communication channel add up to almost 2 hours and 20 minutes per 8-hour workday per participant, which confirms

the importance of the phenomenon reported in previous studies (Robillard & Robillard, 2000)-(Perry et al., 1994-).

6.4.2 Communication Channel Preferences

Figure 6.3 presents the team communication channel frequency distribution during *ad hoc* collaboration.

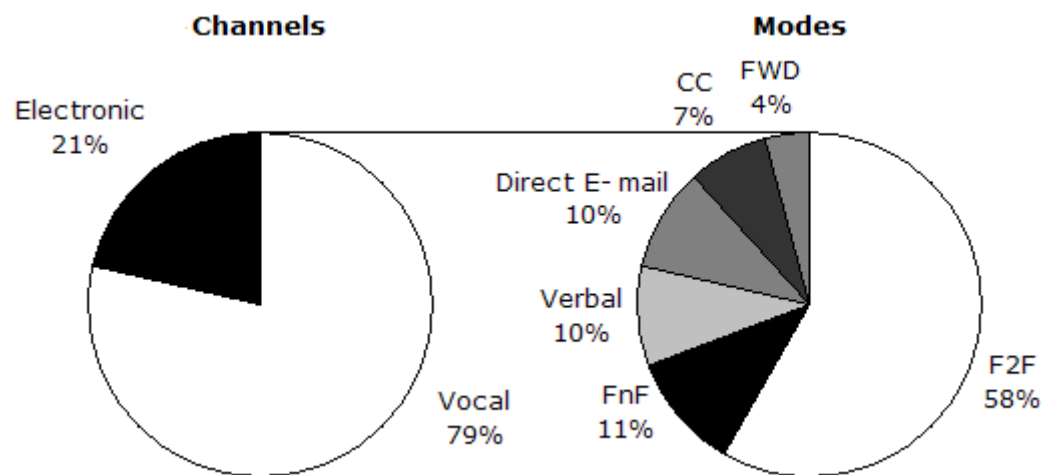


Figure 6.3 Overall communication channels and mode usage during *ad hoc* collaboration

The left-hand chart in Figure 6.3 shows the frequency distribution of channel usage for all the observed *ad hoc* collaborations (other than telephonic interactions, as stated above), during which at least one of the four participants was engaged. Vocal interactions account for 28% of the time spent, as shown in Figure 6.2, and represent 79% of all channel usage, as shown in Figure 6.3. The electronic channel, in the form of e-mails, accounts for 21% of the frequency of *ad hoc* interactions.

The right-hand chart in Figure 6.3 presents the communication channel modes and their frequency distribution for the same *ad hoc* collaborations. These modes were uncovered in the course of the data analysis, and are presented individually, since they illustrate distinctive usage patterns. Accordingly, the vocal communication channel is divided into three modes: face-to-face (*F2F*) interaction, which accounts for 58% of all channel distribution, represents *ad hoc* collaboration in which precisely two participants get together to collaborate, and involves the displacement of one of them to meet the other; *FnF* interaction, which accounts for 11% of channel usage, represents collaboration in which more than two participants get together to collaborate; and verbal interaction, which accounts for 10% of channel usage, represents collaboration where participants remain at their desks. The electronic channel modes all represent communications transmitted via e-mail, however, various types of e-mail can be distinguished: direct e-mails, where an original or reply e-mail is sent to one or more participants, which represent 10% of the overall communication channel distribution; carbon-copy (CC) e-mails, which are the same as direct e-mails, but where some participants are addressed indirectly as “carbon copy” recipients (7%); and, finally, the forwarded (FWD) e-mail, where a previously exchanged e-mail is sent to a third participant (4%). The left-hand chart in Figure 6.3 shows a predominance of the vocal channel over the electronic channel, while the right-hand chart highlights the importance of the *F2F* mode of vocal interaction.

Figure 6.4 presents a subset of the *ad hoc* collaborations included in Figure 6.3, where the communication modes were interchangeable, i.e. the initiator had full flexibility to engage in the interaction in any mode and without any constraint, such

as the physical distance between himself and his interlocutor, the time needed to complete the interaction, etc. According to these criteria, the chart in Figure 6.4 represents 14% of the 514 interactions (404 vocal and 110 electronic).

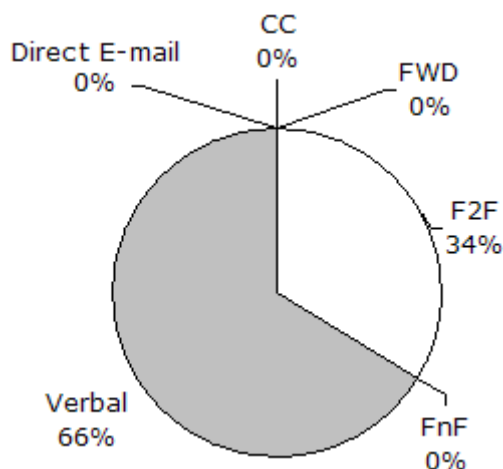


Figure 6.4 Interchangeable *ad hoc* communication mode usage

Based on these criteria, Figure 6.4 reveals a clear preference for verbal interaction. All modes of electronic channel usage drop to 0%, i.e. no participant wanted to communicate by e-mail when it was possible to do so vocally, and verbal mode usage rises to 66% of the interchangeable communication modes, thereby surpassing F2F interactions (34%).

These results are a quantitative indication of the preference of the participants observed for the verbal mode channel when they want to interact with their colleagues. However, when the interactions require more time and/or visual feedback from the interlocutor, F2F appear to be the preferred choice. It is also worth noting that, in almost all circumstances (97% of all e-mails exchanged), e-mail interaction was used when participants had no choice, i.e. to communicate with

a distant stakeholder, to send a file, to forward a previous e-mail to somebody else, or to contact several developers at the same time.

6.4.3 Duration of Communication Channels

Figure 6.5 illustrates the importance of each communication channel mode with respect to duration and usage density.

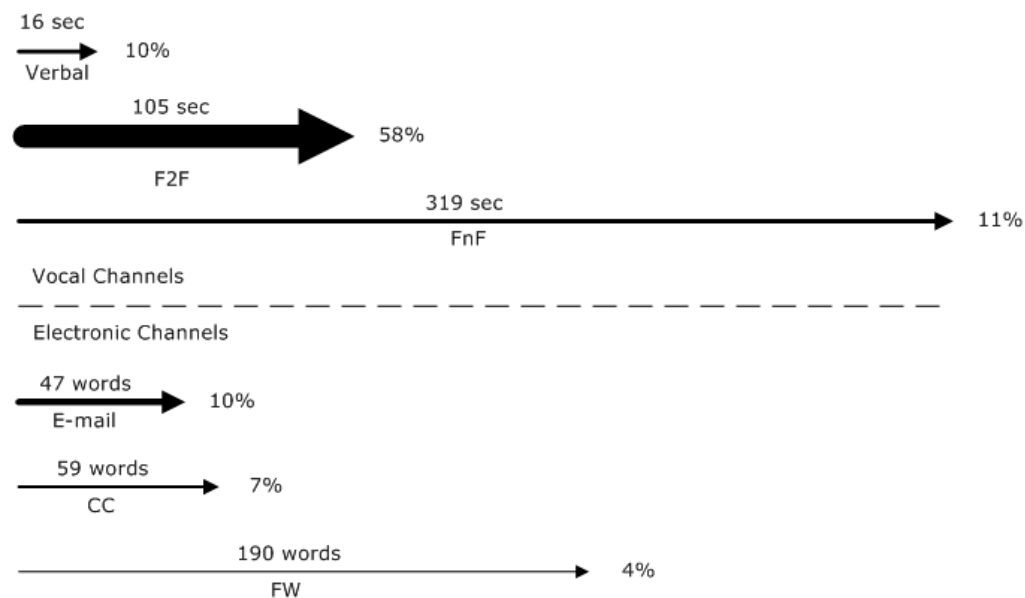


Figure 6.5 *Ad hoc* communication channel mode patterns

In Figure 6.5, each communication channel mode is represented by a vector, the length of which is proportional to its duration, and the size of which is proportional to the distribution of communications through this mode, as revealed in Figure 6.3. The vector length of the vocal modes reflects the median duration of 16 seconds for the verbal mode, 105 seconds for the F2F mode, and 319 seconds for the FnF mode. The vector length for the electronic channel modes are related to their length

in number of words, i.e. a median of 47 words for direct e-mails, 59 words for CC, and 190 words for FWD (FWD message length includes the original message being forwarded). As a result, because of the different metrics used, the length of the vocal modes and e-mails cannot be compared.

Figure 6.5 stresses the different median durations of the vocal modes. Verbal mode is used by neighboring participants wanting to briefly share information. F2F mode is used when a longer exchange is required or when the participants are too far apart to interact verbally. FnF mode generates much longer interactions than F2F mode, as many participants are involved. The following analysis on the content of interactions explains why this is so.

E-mail mode is also characterized by contrasting patterns. Direct e-mail is the most frequent and shortest electronic mode used. CC mode is used less and takes slightly longer, and FWD takes much longer and is used even less.

As shown in Figures 6.4 and 6.5, verbal mode is the preferred communication channel when participants are near one another and the message is very short. Previous published work by the same authors underlines the particular importance of the verbal mode when participants are centrally located within team workspaces (Cherry & Robillard, To be published).

6.4.4 Occurrence of Communication Channels

Figure 6.6 presents the log-normal probability density distribution of *ad hoc* collaborations as a function of their interval duration, which is the time between two consecutive vocal interactions.

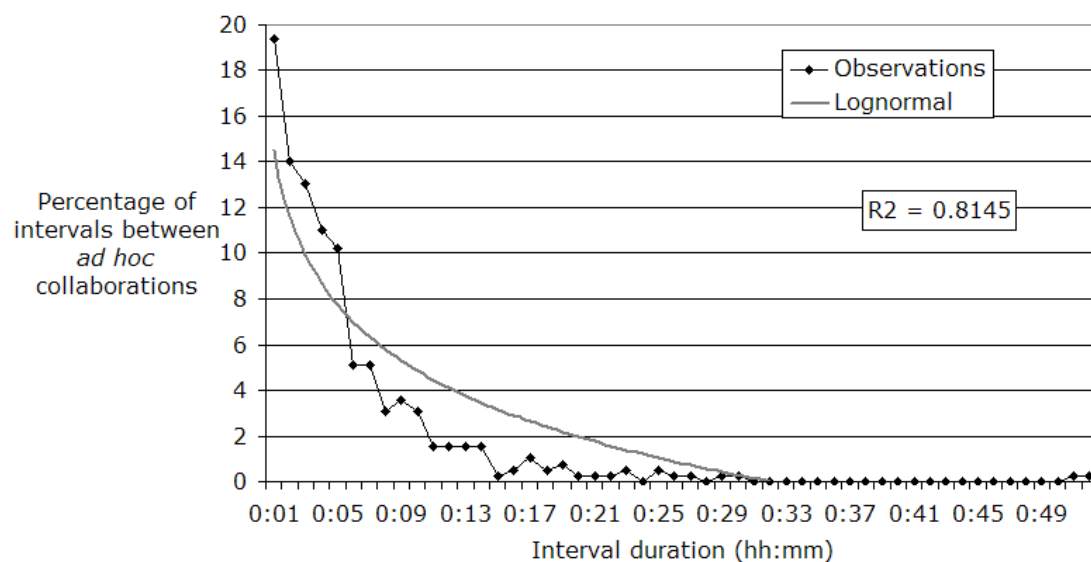


Figure 6.6 Log-normal probability density distribution of vocal *ad hoc* collaborations as function of their interval duration

The probability distribution of vocal *ad hoc* collaborations based on their interval duration strongly follows a log-normal curve, with a coefficient of determination of 0.81. This result suggests that *ad hoc* interactions are independent of one another. *Ad hoc* collaboration seems to occur any time (when) a teammate needs to interact with someone to realize a type of *ad hoc* collaborative activity (why) related to content (what) further detailed.

6.4.5 Participants of Communication Channels

Figure 6.7 presents the distribution of *ad hoc* communication channels as a function of the number of participants involved.

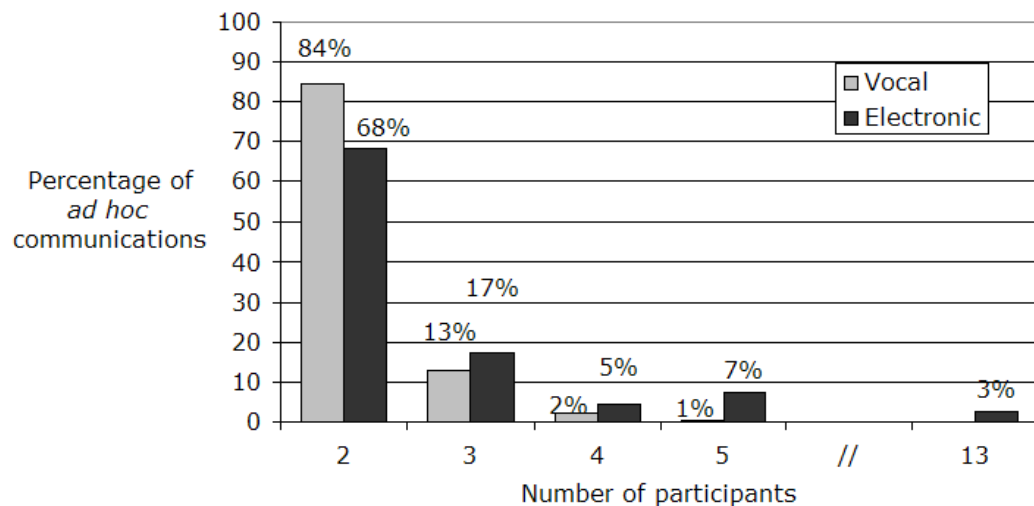


Figure 6.7 Distribution of *ad hoc* communication channels as a function of the number of participants involved

Figure 6.7 shows that most *ad hoc* communication, by means of vocal (84%) or electronic (68%) channels, involved precisely two participants. This result appears to indicate that an *ad hoc* communication is directed to a particular teammate by its initiator. E-mail *ad hoc* communications reached more participants than the vocal channel, which is in line with previous results stating that e-mails are mostly used to reach several participants.

The quantitative results shown in Figure 6.7, supported by qualitative observation, indicate that an *ad hoc* communication is directed from its initiator to the participant, who is expected to respond to it.

6.5 Types and Content of *Ad hoc* Interactions

The following analyses characterize *ad hoc* interactions by identifying the types of activities performed (why), as well as the contents exchanged (what), as introduced by the model illustrated in Figure 6.1.

Types of *ad hoc* communications (why) constitute the key to understanding the cognitive motivations of such collaborations. Five types of *ad hoc* communication were defined over the course of several analysis iterations on the collected data, as follows:

- Cognitive synchronization, which occurs when participants exchange their understanding on a topic to validate their mental representation of the subject being discussed. This exchange of information is always initiated by a request.
 - Example: "Hey Chris, could you please tell me how we should configure this?"
- Information-sharing, which occurs when someone shares information without any prior request from someone else. Unlike *cognitive synchronization*, which occurs in a synchronous mode by a question-and-answer process, information-sharing is asynchronous, in the sense that information is moved to its beneficiary by its initiator.
 - Example: "By the way, Michel, I modified your ABC function in class EFG in order to cover this case."

- Development, which occurs when participants work together on a particular task of the project.
 - o Example: "Let's find the solution to get rid of this problem."
- Coordination, which occurs when the purpose is to schedule meetings or activities, define priorities, or give a status report on an ongoing task.
 - o Example: Martin asks, "When would you be available to meet to discuss that?" Stephan replies, "After lunch." Martin responds, "That's fine with me."
- Irrelevant interaction, which occurs when participants discuss various subjects in a social context, which are not related to the project.
 - o Example: "Hi John! How was your weekend?"

6.5.1 Types of Vocal *Ad hoc* Sequences

Figures 6.8 and 6.9 show the percentages obtained in terms of frequency and duration for each type of *ad hoc* sequence realized by means of the vocal communication channel.

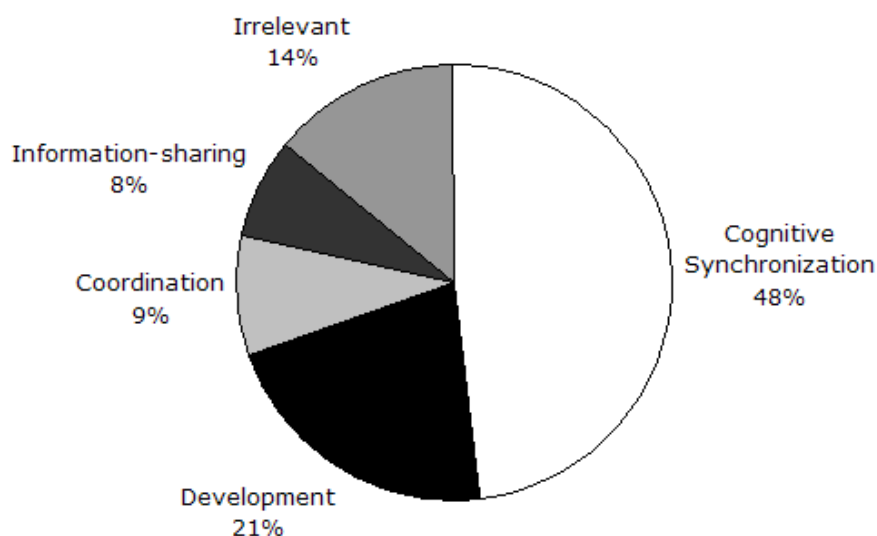


Figure 6.8 Frequency distribution of types of *ad hoc* sequences realized by the vocal communication channel

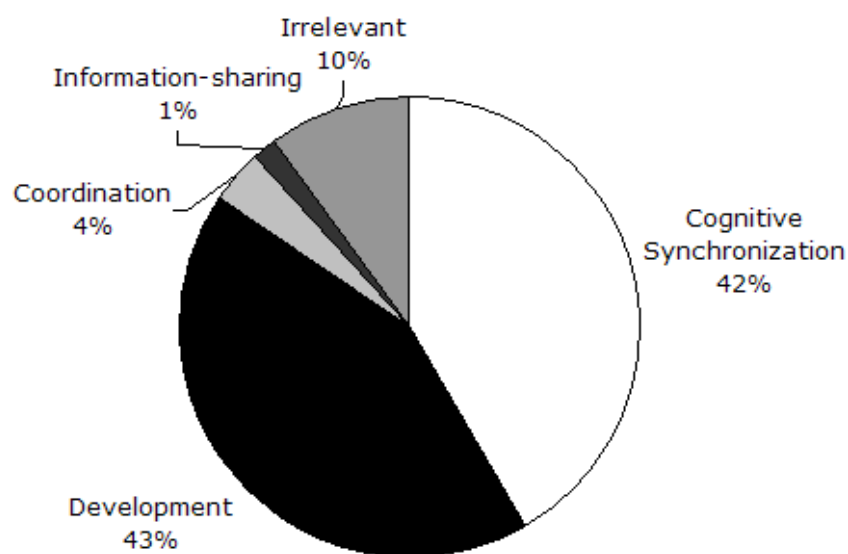


Figure 6.9 Duration distribution of types of *ad hoc* sequences realized by the vocal communication channel

Both figures illustrate that *cognitive synchronization* and *development* are the principal reason why teammates collaborate on an *ad hoc* basis through the vocal communications channel. Figure 6.8 shows that 48% of *ad hoc* sequences are

initiated by participants, to be synchronized with colleagues, in order to obtain the information necessary to pursue their individual tasks. Combined with the result of *information-sharing* (8%), these two categories of synchronous or asynchronous information-sharing represent the majority (56%), in terms of frequency, of *ad hoc* communications realized by participants, and constitute the quantitative demonstration that the intangible and abstract nature of software makes these types of communications necessary.

Development sequences account for 21% of the frequency of *ad hoc* vocal communications, while in time spent the proportion rises to 43%. This type of collaboration is the second most important reason, in terms of frequency, why software developers need to communicate vocally.

Coordination sequences account for 9% of the frequency of the interactions and 4% in terms of time spent. Even though they do not represent a large proportion of the observed *ad hoc* sequence types, coordination sequences nevertheless appear to be important, as previous studies have shown that coordination problems arise in de-located environment settings when *ad hoc* collaborations are limited by distance (Herbsleb & Grinter, 1999).

Irrelevant interaction sequences represent 14% in terms of frequency and 10% in terms of time spent. These social activities may play a role when it comes to team building dynamics and earning the trust of one's teammates. Note that irrelevant sequences have been observed during analyzed working sessions, and so social

activities taking place during lunch or coffee breaks were not included in the collected data.

Figure 6.10 presents the median duration value for each type of *ad hoc* sequence using the vocal communication channel. The median was chosen as the measure for the central tendency because of the highly asymmetric distribution of durations, as indicated by the minimum and maximum values suggested in the value table in Figure 6.10.

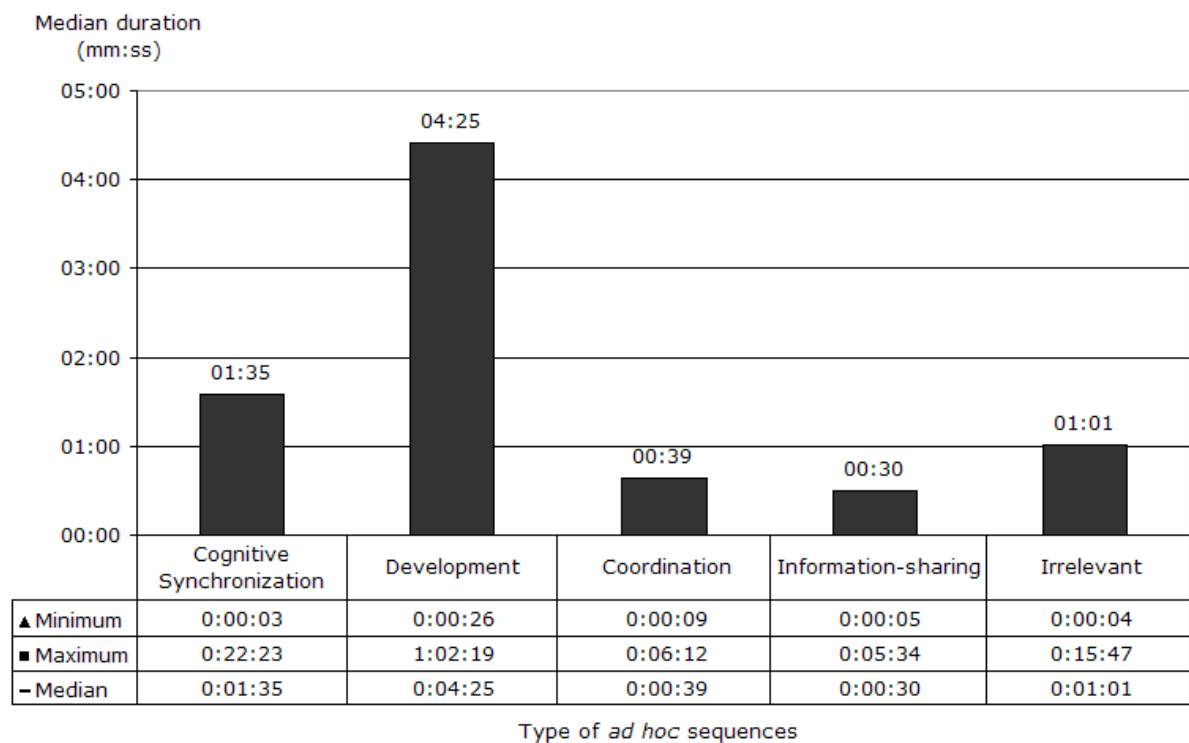


Figure 6.10 Median duration of types of *ad hoc* sequences realized by the vocal communication channel

Two contrasting results emerge from Figure 6.10. On the one hand, the *information-sharing* and *coordination* types of *ad hoc* communication, which showed

the highest percentages in terms of frequency in Figure 6.8, present the shortest median durations in Figure 6.10, with values of 17 and 39 seconds respectively. On the other hand, *development* sequences, which represented 21% of *ad hoc* communications in Figure 6.8 and up to 43% in terms of duration in Figure 6.9, display the highest median duration, with a value of 4 minutes and 25 seconds respectively. *Cognitive synchronization* sequences present a median duration of 1 minute, 35 seconds.

These durations are in line with the intrinsic nature of each of these sequence types. For instance, *cognitive synchronization* activities take longer on average than *information-sharing* activities, even though they both involve the exchange of information. It is, however, safe to assume that these discrepancies are due to the former being a two-way interaction while the latter is not. Moreover, further analysis revealed that very technical topics tend to be discussed during the course of *cognitive synchronization* sequences, thereby explaining the need for the question-and-answer process. The pattern displayed by *coordination* sequences is also in line with their nature, as they occur frequently and are brief, so that teammates can synchronize their activities.

Development sequences last much longer than the other types of sequences, as they involve a work-in-progress on a task to accomplish, which makes their duration variable, depending on the task at hand. The fact that the *development* type of *ad hoc* communication includes problem resolution activities contributes to making this type of sequence the longest.

6.5.2 Types of Electronic *Ad hoc* Communications

Figure 6.11 presents the frequency distribution of types of *ad hoc* sequences realized by e-mails globally, which includes all three modes of electronic communication channel.

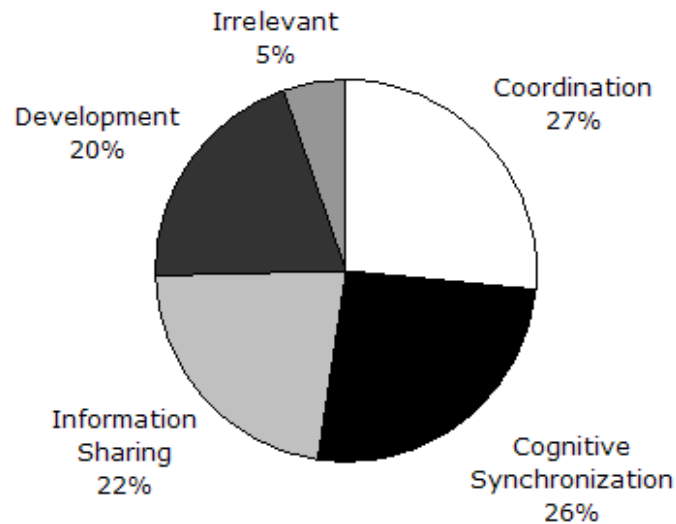


Figure 6.11 Frequency distribution of types of *ad hoc* sequences realized by the electronic communication channel

As outlined in Figure 6.11, the frequency distributions of the *coordination*, *cognitive synchronization*, *information-sharing*, and *development* types of *ad hoc* sequences are almost equivalent, with values ranging from 20 to 27%. This indicates that no type of sequence dominates e-mail exchanges, and so direct e-mails are not used to accomplish any one type of collaboration more than another. However, detailed analysis of forwarded e-mails (FWD) reveals that they mainly involve information-sharing, with value of 64%, which is in line with the nature of such e-mails, as they are sent to secondary recipients to inform them about previous exchanges with

participants. We recall that participants made use of e-mails when use of the vocal communication channel was not possible.

Figure 6.12 combines results from the two preceding sections and reveals the frequency distribution of types of *ad hoc* sequences as a function of communication channels.

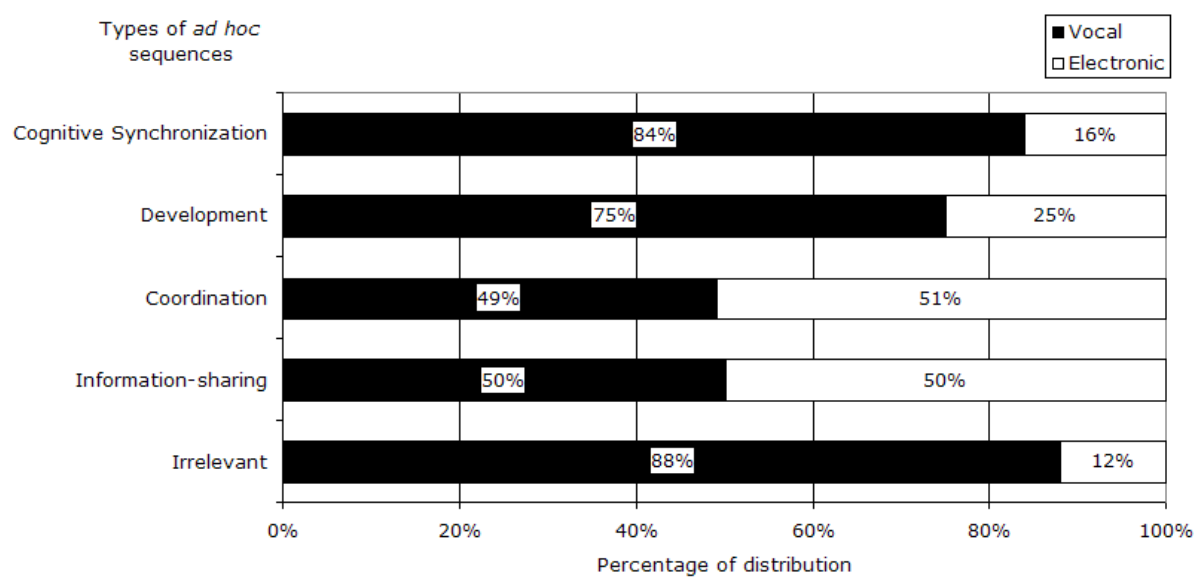


Figure 6.12 Frequency distribution of types of *ad hoc* sequences with respect to the communication channels

The results in Figure 6.12 make it apparent that there are two groups of *ad hoc* sequence types. The first group is composed of *cognitive synchronization*, *development*, and *irrelevant interaction sequences* with values of 84%, 75%, and 88% respectively. Participants seem to prefer to use the vocal channel for these sequences. Moreover, Figure 6.10 reveals that they displayed the longest duration of the five types of sequences. The second group is composed of *information-*

sharing and *coordination* activities, which share the vocal and e-mail channels almost equally. It is also worth noting that Figure 6.10 reveals low durations for both types of *ad hoc* sequence.

6.5.3 Content (what) of *Ad hoc* Sequences

Our analysis of the types of *ad hoc* sequences provides an overview of why participants need to interact by means of *ad hoc* interactions. In this section, we analyze the content (what) exchanged between participants during *ad hoc* sequences. Below are listed the four groups of content that have been defined:

- Specific content group: subjects related to a specific aspect or features of the software product; for instance, a functionality, a specific software component, etc.
 - Example: "John, could you please give me a hand to fix the bug in the XYZ functionality?"
- Technical content group: subjects of general technical interest, which do not relate to a particular aspect of the software built; for instance, a programming concept, a development environment question, a configuration management issue, etc.
 - Example: "Hey Michel. Where is the setting to disable the compilation of these files in the IDE?"
- Organizational content group: subjects related to the development process, or to resources, training, meetings, etc.

- o Example: "Martin, can you tell me when the integration test phase has been planned to begin?"
- Irrelevant content group: subjects which do not concern the project.
 - o Example: "Hey Martin! How about pizza for lunch?"

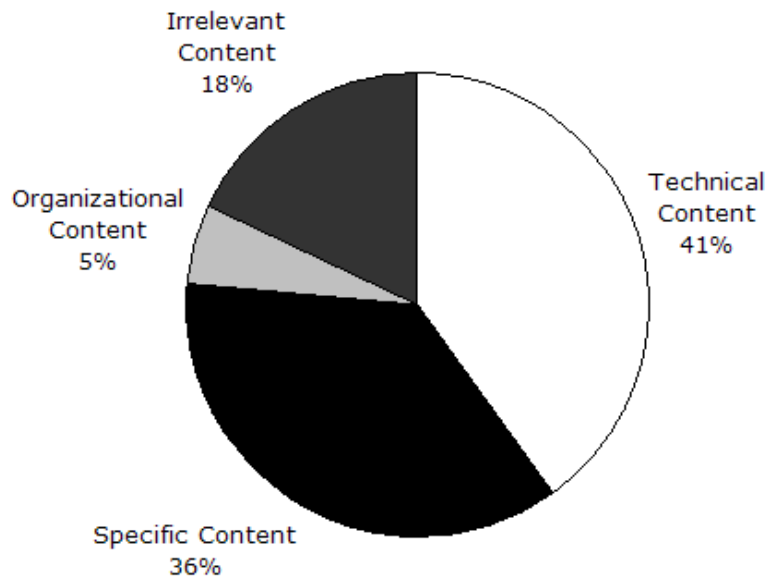


Figure 6.13 Frequency distribution of *ad hoc* sequence content by means of the vocal communication channel

Figure 6.13 shows that vocal *ad hoc* sequences are mostly concerned with technical (41%) and specific content (36%). Accordingly, Figure 6.13 reveals that the majority (77%) of *ad hoc* sequences are task-centric. This is in line with the types of *ad hoc* sequences revealed previously, to the effect that *ad hoc* communications mainly involve information-sharing (cognitive synchronization and information-sharing) and development activities, two types of sequences which are also essentially task-centric. Organizational content, by contrast, relates to all the peripheral issues (excluding those associated with the actual building of the software and the know-how required to do so), while the coordination type of

sequence represents management activities on any issue. This content occupies rather a lower proportion of the distribution, at 5%.

Figure 6.14 shows the relationships between the types of vocal *ad hoc* sequence (why) described previously and their content (what).

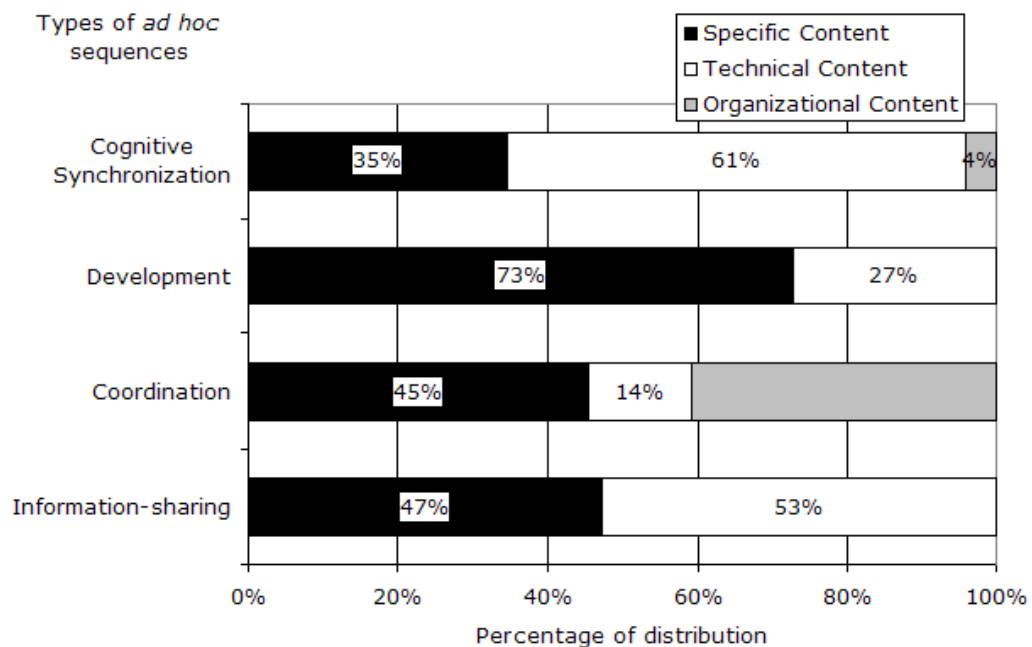


Figure 6.14 Distribution of content as a function of the type of vocal *ad hoc* sequence

Figure 6.14 shows that, on the one hand, cognitive synchronization sequence types appear to be mainly concerned with technical content (61%) and specific content (35%), while the last 4% is related to organizational content, which concerned questions raised about a reorganization in a department with whom participants had to deal. On the other hand, and as expected, development sequence types are related to specific content (73%). In the middle of the spectrum, information-

sharing sequence types are divided almost equally between specific content (47%) and technical content (53%).

Figure 6.14, combined with the results in Figures 6.8 and 6.9, show that many interactions involve the exchange of information about general aspects of software development and the development environment, and are not necessarily related directly to the software product to be built.

6.5.4 Focus of *Ad hoc* Communications

Figure 6.15 presents the number of sequences per interaction. We recall that sequences are the smallest analyzed unit within interactions, and take the form of a block of one or more exchanges between participants relating to the same content (what) or orientated towards the same type of activity (why).

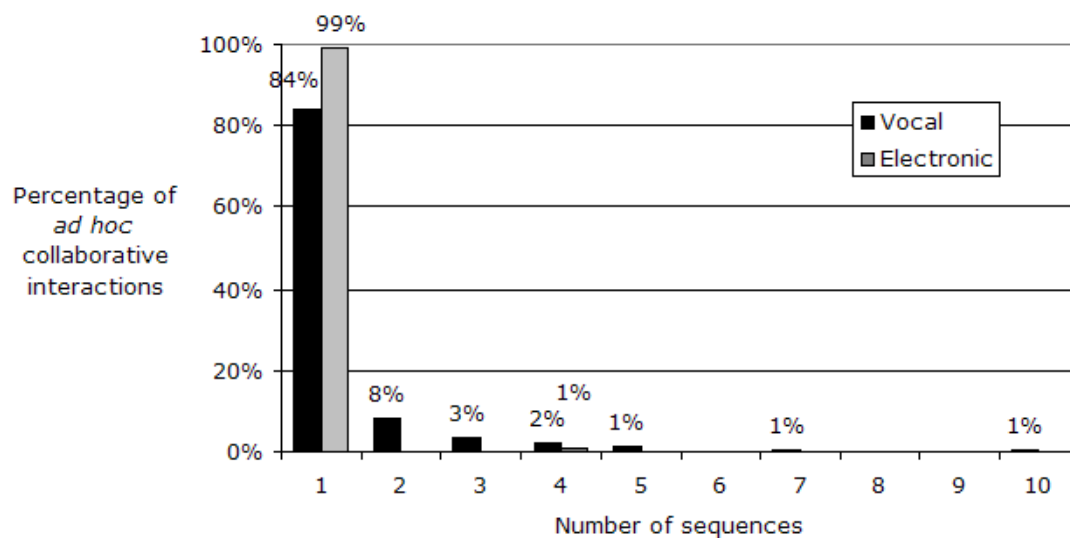


Figure 6.15 Percentage of *ad hoc* collaborations per communication channel as a function of the number of sequences per interaction

Figure 6.15 shows that 99% of the interactions on the electronic communication channel are composed of one interaction of a single sequence of the (why) type of communication about the same content (what). The vocal *ad hoc* communication channel presents the same pattern, with most interactions composed of a single sequence, like the electronic channel, but with a few more interactions composed of two, three, four, five, seven, or even ten sequences. Detailed analysis reveals that all vocal interactions (100%) with more than two sequences involve more than two participants, and so are realized as FnF interactions.

These results indicate that *ad hoc* interactions are initiated for a specific reason (why), are focused on a specific subject (what), and, as shown in Figure 6.7, target a specific participant (who). However, when more than two participants are involved in *ad hoc* interactions, they tend to be less focused on one purpose and address several issues.

6.5.5 Relevance Level of *Ad hoc* Sequences

Some authors are concerned with the relevance of the information that is exchanged during these undocumented *ad hoc* interaction sequences (LaToza et al., 2006). However, it is important that some rationale that might be important for product development not be lost. We propose a point of view for evaluating the relevance of content based on the usefulness of the type of information exchanged. All sequence contents observed in this study have been categorized according to the following three usefulness relevance levels:

- *Knowing* relevance level: identifies factual contents which are relevant when the interaction occurs. This is a type of content which is relevant for a given participant, for a specific reason, in a well-defined context, and not likely to be relevant subsequently.
 - o Example: "Hey John, did you release class ABC?"
- *Know-how* relevance level: identifies contents which are related to the skills required to build the product, such as technical ability and expertise, and not the product itself.
 - o Example: "Hey Martin, how should we deal with the new branch in the source control?"
- *Knowledge* relevance level: identifies contents which are related to the product itself, its design and implementation, as well as to the business application and rules that it implements.
 - o Example: "Hey Chris, could you please tell me which business rules are handled in function XYZ?"

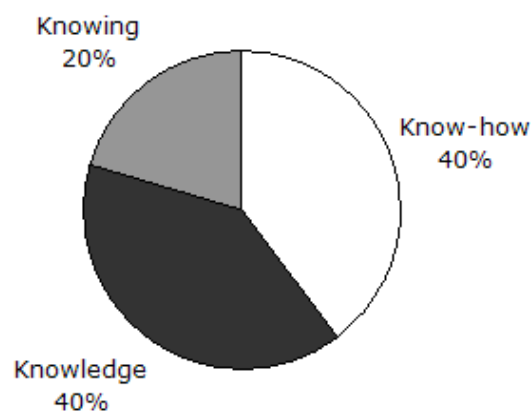


Figure 6.16 Frequency distribution of vocal *ad hoc* collaboration relevance levels

Figure 6.16 reveals that up to 40% of the content exchanged using the vocal communication channel is related to the *know-how* relevance level, while the same percentage (40%) is observed for the *knowledge* relevance level. This indicates that participants collaborate in this particular study setting as often about the technicalities of how to build the product as those of the product features themselves. These two categories represent content that could potentially be useful over the long term, since there is a high probability that it could be requested again and be the cause of other interactions, since no trace of this information exists outside the participant's mind. LaToza (LaToza et al., 2006) made the same point: developers may invest a great of effort in recovering implicit knowledge by exploring code and interrupting teammates, and this knowledge is only saved in their memory. The *knowing* content level, which accounts for 20% of the distribution, identifies factual content, with short-term usefulness, normally when information is needed to continue an individual task and is not useful subsequently.

Figure 6.17 illustrates the frequency distribution of relevance levels as a function of the types of *ad hoc* sequences described and observed in Figures 6.8 and 6.9.

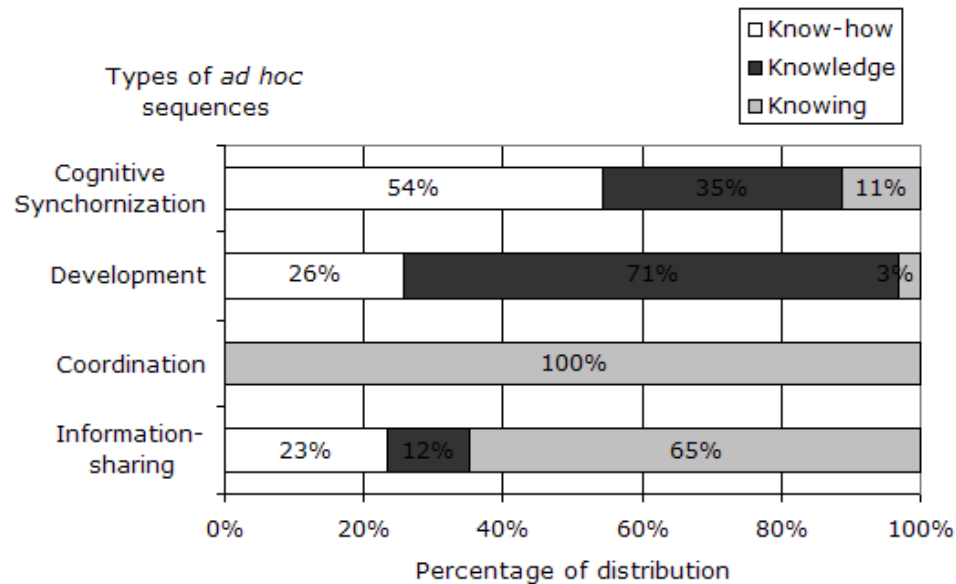


Figure 6.17 Frequency distribution of relevance levels as a function of the type of vocal *ad hoc* sequences

Figure 6.17 shows that the *know-how* content level accounts for half (54%) of *cognitive synchronization ad hoc* sequences. These include, for instance, exchange of information about an integrated development environment (IDE), system configuration, software components, or libraries used in the construction of the software, etc. The *knowledge* relevance level accounts for a third (35%) of the cognitive synchronization type of sequence. *Cognitive synchronization* is more concerned with issues related the construction of the software than to those related to the software product. Only a weak percentage (11%) of *cognitive synchronization ad hoc* sequences involve the *knowing* relevance level. These data show that *cognitive synchronization* is mostly performed to synchronize participants in the working environment, which is shown by the two third (65%) of sequences involving the *know-how* and *knowing* relevance levels.

Up to three quarter (71%) of *development ad hoc* collaborative activities are composed of knowledge related to the product, and only a quarter (26%) by know-how content relevance. This is in line with the definition of the *development* type of *ad hoc* communication, which is observed when two or more participants work on a particular task of the project, and so on a particular artifact of the software product. Again, only a trace (3%) of this type of *ad hoc* communication involves the *knowing* relevance level of content, meaning that a large proportion of them may be relevant over time.

Coordination ad hoc collaborative activities are composed entirely (100%) of *knowing* relevance level sequences, meaning that the content of these sequences is not relevant over time and only allows participants to coordinate their activities in a timely manner.

The *information-sharing* type of *ad hoc* collaborative activity mainly comprises the *knowing* level of content relevance, with a proportion of two third (65%), thus representing a type of sequence with potentially low relevance over time.

While the results in Figure 6.17 reveal that some particular *ad hoc* sequences appear likely to include content that is relevant over time, they also identify particular types of sequences where content is more likely to involve implicit knowledge about the product, and other content that is related rather to the skills required to build software in this organization.

Figure 6.18 presents the frequency distribution of relevance types as a function of the vocal *ad hoc* communication modes identified and presented in Figure 6.3.

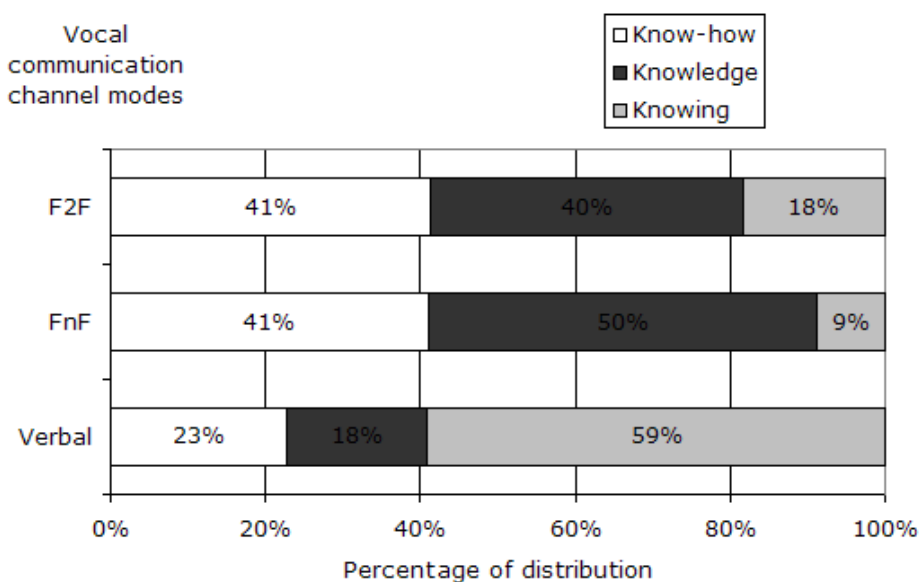


Figure 6.18 Frequency distribution of relevance types as a function of vocal *ad hoc* communication modes

Figure 6.18 shows that the verbal interaction mode, which is mostly based on the *knowing* relevance level sequences, is noticeably different from the other two modes, which are split almost half and half between *knowledge* and *know-how* sequence relevance levels.

Figure 6.18 shows that, for instance, 41% of *ad hoc* communications realized by the F2F mode are at the *know-how* level of relevance, while *knowledge* relevance represents 40% of that relevance level. FnF presents somewhat the same pattern, with 50% and 41% of the *knowledge* and *know-how* types of content relevance respectively. Both the F2F and FnF communication modes revealed low proportions (18% and 9% respectively) of the *knowing* type of content relevance. However, the

verbal communication mode illustrates the opposite pattern, by presenting a large proportion of the *knowing* type of content relevance at 59%, and noticeably lower proportions of *know-how* (23%) and *knowledge* (18%) relevance.

Figure 6.18 indicates that the verbal communication mode mainly involves short-term relevant content, while F2F communication modes involve potentially large proportions of content that is relevant over time.

6.6 Discussion

Previous publications by the same authors have revealed the network patterns of participants (who) taking formal or informal roles in *ad hoc* collaboration (Cherry & Robillard, 2008). This paper illustrates the usage patterns of team *ad hoc* collaborations, i.e. the communication channels used (how), the number of stakeholders engaged (who), and the occurrence of these collaborations (when), and further details the content of *ad hoc* collaborations with regard to the types of sequences (why) realized on particular content (what) and the level of relevance.

As the central objective of this paper, the patterns and content of *ad hoc* collaborations outlined in previous sections provide a better understanding of the *raison d'être* of these informal collaborative activities and of how the collaboration process can be enhanced within the team. Furthermore, this understanding also provides the first elements of response to address contradictory ideas found in the literature and reported earlier.

6.6.1 Opportunistic Nature of *Ad hoc* Collaborations

One of the major outcomes of the results presented in this paper is that they shed light on the *opportunistic* nature of *ad hoc* communication. First, we have seen that preference seems to drive the choice of some communication channels when the time comes to initiate an interaction with a teammate. In fact, the shortest communication path is preferred when there are options available. Results then revealed that *ad hoc* communication occurs at an opportune time, that is, when a teammate needs the collaboration of his colleagues. Moreover, *ad hoc* communication is directed toward a single teammate who is likely to respond to the need for collaboration.

Cognitive synchronization appears to be the main reason why teammates need to collaborate. With the patterns described above, it turns out that the need for information to progress on an individual task can occur at any time, and developers address this need at the opportune time, that is, when an answer is required to complete the task. *Development* sequences present the same intrinsic motivation when questions come up about what to do and about how or when problems appear, in that developers seek the help of colleagues to obtain missing information, to ask for advice, or for guidelines to pursue their tasks. The *coordination* type of *ad hoc* sequence also seems important to the organization of the effort deployed by the team, to synchronize the activities of teammates on the tasks to tackle, and to plan further activities. Even though coordination is not the predominant type of *ad hoc* communication in terms of frequency and time spent, it appears that coordination activities, however brief, are vital to maintaining team

activity and to synchronization, as well as to avoiding deadlocks or confusion in task organization.

The topics of exchanges yielded similar, quite clear tendencies, with up to three quarter (77%) of topics (36% on specific subjects and 41% on technical aspects) oriented to the tasks at hand. Finally, results revealed that *ad hoc* collaborative interactions, when they occur, tend to be highly targeted to one particular topic (what) to realize one specific type of activity (why), which corroborates our finding that these interactions emerge to address an inherent need.

Based on the patterns described above, *ad hoc* collaborations seem to originate noticeably from a natural, unconscious, and *ad hoc* opportunistic process that takes place during software development activities. As discussed by Robillard (Robillard, 2005), an *opportunistic* process, in contrast to a *systematic* process, where all the knowledge required to complete a task is available so that a well-structured plan can be followed, is defined by an incremental process in which knowledge is gathered as opportunities present themselves, and depends on the cognitive availability of the necessary information. This process has been observed, in the software development settings studied, in the form of the *ad hoc* collaboration of teammates to gather, in a *just-in-time* manner, the information they need to accomplish the task at hand. Of course, not all *ad hoc* types of sequences, such as *information-sharing* and *irrelevant interaction* activities, originate from such an opportunistic process. However, a large majority, such as the *cognitive synchronization*, *development*, and *coordination* sequence types are likely to follow opportunistic patterns.

This key outcome has important theoretical, and hence practical impacts. Authors in the literature have stressed, as reported previously, the notion of interruptions (Jett & George, 2003) and their negative impact on the flow and continuity of an individual's work. Some have proposed limiting them; for instance, by adopting rules such as *Bring back the door*, suggested by DeMarco and Lister (DeMarco & Lister, 1999). Nevertheless, the results presented above not only raise concerns about the interpretation of *ad hoc* communications strictly as interruptions, but even lead us to believe that interruptions may occur as a manifestation of inherent needs to be fulfilled. Moreover, latency in response to those needs could prove to be detrimental to an individual's work, as well as to the team dynamics.

Working in an opportunistic mode, software developers evolve as long as their knowledge is sufficient to accomplish the tasks at hand. Erecting barriers to the opportunistic process of seeking the required information in *just-in-time* fashion seems, in light of the results presented above, contrary to the natural cognitive way software developers work in their everyday abstract environment. For instance, limiting the opportunistic and *just-in-time* coordination activities in any way could cause problems, as already observed by Herbsleb and Grinter in de-located contexts where distance poses such a challenge (Herbsleb & Grinter, 1999).

6.6.2 Task-centric Content of *Ad hoc* Communications

Results on the content (what) of *ad hoc* communications have uncovered two contrasting content types exchanged during *ad hoc* interactions. On the one hand,

technical content was exchanged in up to 41% of *ad hoc* communications, meaning that a large number of sequences did not relate specifically to the software to be built, but to general *know-how* matters. This might explain why software developers need to interact on an *ad hoc* basis, since artifacts prescribed by software processes formalize required *knowledge* related to what software should be built and how, but does not, of course, address *know-how* content that is not suitable for this kind of artifact. Furthermore, according to the content relevance level, a combined portion of 40% of *know-how* content was observed. This led us to believe that recurring interactions regarding these questions are to anticipate, and constitute a great opportunity for collaboration optimization.

On the other hand, the results reveal that a third (36%) of the exchange topics concern specific aspects related to the software to be built, as the results on relevance levels that revealed as well 40% of knowledge-related content. Studies on the information needs of software developers identified the search for a design and code rationale (Ko et al., 2007; LaToza et al., 2006) as an important source of *ad hoc* interruptions. The *knowledge* relevance level categorized in this study is a sub category of *ad hoc* communication, where exchanges concern rationale issues. As pointed out by previous authors, the degree to which diverse specific contents of the software to be built are discussed raises questions about the adequacy and accuracy of the array of artifacts prescribed by software processes (Ko et al., 2007).

6.6.3 Reserved Use of E-Mails in Co-Located Settings

Another major outcome of this research is the observation that the use of e-mails in a co-located context is limited almost entirely to instances where participants were obliged to use this channel to transmit attached files or to communicate with distant and/or numerous stakeholders. This raises questions about the adequacy of this asynchronous channel for meeting *opportunistic ad hoc* communication needs, a channel that is by nature quietly different from vocal interactions, which are synchronous and so provide instant feedback, offer the possibility of reacting promptly, thereby facilitating exchanges. While this does not constitute a major obstacle to *ad hoc* communications in a co-located context, such as the one featured in this study, where teammates are free to interact vocally, it could prove to be a real barrier in a de-located context, as observed by Herbsleb and Grinter (Herbsleb & Grinter, 1999). More research is needed to investigate the use of e-mail in de-located contexts and the impact of instant messaging technologies on *ad hoc* collaboration in both co-located and de-located contexts.

6.6.4 Research Concerns

6.6.4.1 Ethical Issues

We considered the ethical implications of this research early on, at the planning stage, and ensured that all subjects and the hosting organization understood their rights and responsibilities before they agreed to participate (Singer & Vinson, 2002). Every means possible was employed to guarantee the ethical use of the recordings. An Ethics Certificate was issued for this research by an independent committee of

the École Polytechnique de Montréal, which is mandated to supervise research undertaken with human subjects. All the individuals involved in our study were duly informed that their work sessions would be audio-visually recorded, as well as of the nature of the study. They all signed the letter of agreement required for certification. Ethical issues were handled according to the established Canadian policies for research involving human subjects (NSERC, 2005).

6.6.4.2 Scientific Value

The scientific value of this research has two components: the importance of the research topic and the validity of the experimental results. The importance of the research topic is threefold: the human and collaborative aspects of software engineering, software process, and global software development.

The human and collaborative aspects of software engineering constitute the major factor at issue here, and understanding these aspects in the context of software development is crucial to understanding how practices could be improved. The emerging Agile philosophies rely mostly on human interaction, and a better understanding of its purposes can shed light on new practices more suited to the various software process approaches. Global software development involves a de-located team, where easy verbal and F2F interactions via electronic means continue to occur. Language, culture, and time lags are important parameters. Our study shows some of the parameters involved in the usual *ad hoc* interactions among co-located team members. This study can serve as a basis for a more in-depth study of the impact of *ad hoc* interaction via electronic channels.

The validity of the observed results mostly relies on the fact that they faithfully and reliably represent reality. This study was not an experiment, where the various parameters could be controlled, and this paper reports observations performed in a real professional environment in the course of carrying on day-to-day business. The salient outcome of these observations, regardless of the specific setup of the organization, is a better understanding of the *raison d'être* of *ad hoc* collaboration and how it takes place.

6.6.4.3 Study Limitations

One limitation of this study is that it was conducted in the form of a case study, and so conclusions ensuing from the current research cannot be generalized to all software development settings. However, owing to the characteristics that the featured settings have in common with the software development environments that can be encountered in the industry, we can assume that the outcomes of this study can be applied to a broader set of organizations.

Moreover, the types of *ad hoc* communication defined in the framework of this research have been inducted and observed in a maintenance context instead of a development context. Since maintenance contexts, where existing software is enhanced, predominate in software engineering settings, it is safe to assume that the results of this research are applicable to a broad set of contexts. However, it is possible that the defined set of *ad hoc* communication types is not exhaustive and must be extended to cover development environments specifically.

Finally, the method of audio-video recording chosen for collecting some of the data for this research poses a number of challenges, such as the background noise emerging from an open work space hosting hundreds of developers. It has several advantages, however: it can be reviewed as often as needed, and it is considered by participants to be less intrusive than having an observer take notes on their activities.

6.7 Conclusion

Through this research, it has been possible to shed some light on the main aspects of *ad hoc* collaboration in terms of the communication mechanisms (how) in which they are conducted, but more importantly their motivation (why) and the content (what) exchanged during those activities. This better understanding has revealed the opportunistic nature of *ad hoc* collaboration, which constitutes the cornerstone for further theories and research about the phenomenon. It also paves the way for the introduction of new practices, in co-located as well as de-located contexts, to improve *ad hoc* collaboration. Further research will be required to test their applicability. Additional studies will also be required on the clearly opportunistic work method that developers exhibit, as opposed to the systematic approach, and, from a broader perspective, the impact of this cognitive mechanism on software processes. In short, we believe that the cognitive aspects of software engineering will continue to provide research opportunities for years to come.

6.8 Acknowledgments

This research would not have been possible without the agreement of the company in which it was conducted, and without the generous participation and patience of the software development team members from whom the data were collected. To all these people, we extend our grateful thanks. Thanks also to Michel Martin for his generous participation in the inter-coder agreement phase.

CHAPITRE 7

SOMMAIRE DES CONTRIBUTIONS

La recherche présentée au cours du présent document a permis de donner fruit a de nombreuses contributions à plusieurs points de vue et en voici le sommaire.

7.1 Apports méthodologiques

En regard à l'aspect méthodologique, cette recherche constitue un exemple d'étude empirique sur laquelle peuvent se baser plusieurs chercheurs dans l'entreprise de recherches ultérieures visant à l'approfondissement des connaissances en regard aux facteurs humains impliqués en génie logiciel et ce, à plusieurs égards.

D'abord, l'approche par observation participante exposée dans ce document, particulièrement l'utilisation des enregistrements audio-vidéo faite dans le cadre de cette étude ainsi que les motivations exposées pour en faire l'usage contrairement à la prise de notes sur le terrain, tel qu'il est souvent coutume de faire dans le contexte de recherche en observation participante et exploratoire, permet de promouvoir le recours à cette méthode de collecte de données, non plus seulement dans certains domaines précis du génie logiciel, tel que l'étude d'interfaces homme-machine, mais également dans toutes sphères de recherches portant sur les aspects humains en génie logiciel.

De plus, l'analyse des données à l'aide de l'*Exploratory Sequential Data Analysis* (ESDA) telle que proposée par Fisher et Sanderson (Sanderson & Fisher, 1994; Fisher & Sanderson, 1993) a permis de rappeler que l'analyse des données dans le cadre de recherches portant sur les aspects humains en génie logiciel peut être abordée par l'entremise de trois approches différentes et qu'un amalgame de ces approches, tel que celui imaginé dans le cadre de cette recherche, peut être adopté afin d'adapter la méthodologie de recherche à l'objet étudié ainsi qu'au terrain particulier sur lequel se déroule l'étude. Aussi, cette recherche a permis, d'une part, d'apporter un autre exemple concret démontrant comment est-ce que le processus de transformation de données qualitatives en données quantitatives peut s'avérer utile lors de recherches empiriques comme celle-ci afin d'être en mesure de tirer le plein potentiel des données recueillies et d'approfondir leur analyse.

Par ailleurs, la technique d'encodage ne pouvant pas toujours être réalisée à l'abri des pièges de la subjectivité, une approche de mesure de subjectivité a été décrite afin de pouvoir constater l'ampleur de cette subjectivité. Toutefois, même si la subjectivité est souvent considérée comme étant un élément à éviter en recherche, divers arguments théoriques ont été rappelés au cours du chapitre de stratégie de recherche pour soutenir que non seulement la subjectivité dans certaines situations ne constitue pas nécessairement un biais méthodologique, mais elle peut être dans certaine situation souhaitable afin d'être en mesure d'analyser les données avec plus de justesse et pouvoir en tirer des conclusions qui auront une validité accrue sur le terrain étudié. Cet aspect méthodologique aurait par ailleurs tout intérêt à être davantage considéré ou du moins débattu, en génie logiciel.

En somme, l'apport méthodologique de cette étude est qu'elle a permis d'attirer l'attention sur différentes approches méthodologiques qui étaient jusqu'à maintenant encore très peu adoptées dans le cadre de recherche empirique en génie logiciel. L'application de ces méthodes à l'étude du travail collaboratif *ad hoc* a été démontrée, études qui doivent être poursuivies, notamment sous le regard de la tradition cognitiviste tel que proposé par Fisher et Sanderson (Sanderson & Fisher, 1994) afin de détailler le modèle théorique du phénomène étudié entrepris dans le cadre de cette recherche, et qui reste à parfaire.

7.2 Apports théoriques

Bien que l'apport méthodologique de cette recherche à la science du génie logiciel soit appréciable, n'en demeure pas moins que la contribution majeure de cette recherche est d'ordre théorique. Cette recherche a permis de décrire, de par les résultats obtenus et dévoilés au cours des trois articles exposés dans le présent document, un phénomène du génie logiciel qui, bien qu'important en terme de temps et reconnu autant par les chercheurs que par les praticiens, était auparavant encore profondément méconnu de tous.

Ainsi, répondant à la question initiale de recherche qui visait, d'une part, à caractériser le travail collaboratif *ad hoc*, cette étude a de la sorte non seulement permis la caractérisation du travail collaboratif *ad hoc* réalisé au sein de l'équipe de concepteurs de logiciels et la description des patrons et du contenu des échanges impliqués, mais elle a également permis l'élaboration d'un modèle théorique sur quatre principaux plans à savoir les acteurs impliqués (qui), les canaux de

communications utilisés (comment), les types d'activités collaboratives *ad hoc* réalisées (pourquoi) et le contenu échangé (quoi). Pour rappeler les principales observations notons l'émergence de différents acteurs (qui) dont le rôle formel ou informel occupé au sein de l'équipe explique en partie l'influence jouée et l'important apport de ces acteurs lors de certains types d'activités (pourquoi). Le rôle de la recrue (qui) permet également de dénoter l'apparente quête de collaboration de la part de ses collègues afin d'amasser les informations et connaissances nécessaires à son évolution au sein de l'équipe. Les résultats permettent également de jeter un éclairage sur l'usage des canaux de communications (comment) et l'apparente préférence quant aux canaux vocaux, plus particulièrement au mode verbal, par rapport aux canaux électroniques voire téléphoniques pour la tenue d'activités collaboratives *ad hoc*.

D'autre part, les analyses, discussions et hypothèses induites et présentées au cours des articles rapportés ont permis de proposer des explications possibles quant à la seconde partie de la question de recherche initiale qui cherchait à expliquer pourquoi le travail collaboratif *ad hoc* pouvait occuper autant de temps de façon informelle en dépit de la nature formelle des processus de génie logiciel. Non seulement le modèle formé des patrons et contenus présentés a permis de déceler la nature opportuniste ou «juste-à-temps» des activités collaboratives *ad hoc* qui prennent majoritairement forme (pourquoi) en synchronisation cognitive ou activités de développement, mais il a également permis d'illustrer la nature du contenu échangé qui comporte une très large part de «savoir-faire», types d'informations et connaissances qui ne sont normalement pas tenus en compte au sein des processus de génie logiciel. Aussi, il est apparu que les «connaissances»

sur le produit développé normalement colligées par le biais d'artéfacts prescrits par ces même processus de génie logiciel, sont largement véhiculées lors d'activités collaboratives *ad hoc*. Une étude plus approfondie sur cette découverte s'avèrerait nécessaire afin d'en déterminer la cause.

De l'apport théorique de cette étude, non seulement de nombreux débouchés pratiques peuvent en découler, mais également de nombreuses recherches peuvent en dériver. Par ailleurs, ce modèle ayant été élaboré à partir de concepts qui ne sont pas nécessairement propres au terrain étudié, celui-ci pourrait donc être transposé dans un contexte beaucoup plus large que le cas étudié. Offrant ainsi aux éventuels chercheurs du domaine les premiers points de repères théoriques à partir desquels ceux-ci peuvent baser leurs recherches, cette étude exploratoire permet maintenant, d'une part, d'orienter les travaux futurs dans ce domaine méritant bon nombre d'éclaircissements, mais d'une autre part, dévoile du même coup tout un pan d'opportunités de recherches pouvant être entreprises.

7.3 Apports à la pratique

Même si cette recherche s'avère essentiellement fondamentale, celle-ci comporte néanmoins certaines visées pratiques, le but étant, au-delà de comprendre le phénomène à l'étude, d'améliorer ultimement la pratique du génie logiciel. Découlant donc des percées théoriques que cette étude a pu apporter, les retombées pratiques de cette recherche sont également substantielles.

D'une part, les observations réalisées permettent maintenant d'identifier des pratiques qui puissent être contraire à la nature inhérente des activités collaboratives *ad hoc* et ainsi répondre aux divergences de point de vue trouvées dans la littérature sur le sujet. Elles permettent également la définition de nouvelles pratiques plus adaptées à la réalité empirique des besoins en terme de collaboration au sein d'une équipe de concepteurs de logiciel. Bien que cette recherche n'aie pas comme objectif premier la définition de ces nouvelles pratiques qui permettront, il est permis de l'espérer, de rendre le travail collaboratif plus efficace en génie logiciel, elle aura néanmoins permis d'en ébaucher certaines au cours des articles, notamment en ce qui a trait à l'ergonomie des espaces de travail et à la disposition des coéquipiers au sein de celle-ci, à la gestion de projet prenant compte des rôles formels et informels des différents acteurs du projet, à la facilitation des activités collaboratives *ad hoc* de nature opportuniste plutôt que leur limitation, etc.

Plusieurs autres pratiques pouvant également être promulguées n'ont pas été évoquées au cours des articles, pratiques telles que la prescription de formes d'artéfacts en vue de conserver le «savoir-faire» échangé lors d'activités collaboratives *ad hoc* de même que la tenue de réunions ponctuelles pour gérer des événements tout aussi ponctuels qui peuvent engendrer de multiples interactions *ad hoc*. De telles pratiques n'ont pas été validées dans le cadre de cette recherche, mais celles-ci pourront être confirmées au cours de recherches ultérieures.

Par ailleurs, étant donné que le travail collaboratif *ad hoc* constitue d'abord et avant tout une activité communicative, une meilleure compréhension de ce type d'activités et de leur contenu permet maintenant de définir des requis en terme

d'outils de communication à disposer tant au sein d'une équipe œuvrant sur le même lieu physique, qu'une équipe virtuelle distribuée géographiquement. La fréquence et la durée des différents types d'activités collaboratives *ad hoc* recensées, la nature du contenu échangé, le nombre et le rôle des intervenants impliqués et surtout, les canaux et modes de communications privilégiés sont tous des intrants à la définition de tels besoins. Encore une fois, cette recherche n'aura pas tenté de définir une liste exhaustive de ces besoins, mais elle en aura permis l'ébauche des grandes lignes.

CONCLUSION

Ce document décrit les résultats d'une étude empirique qui visaient à caractériser et décrire le contenu du travail collaboratif *ad hoc* réalisé au sein d'une équipe de concepteurs de logiciels au cours d'un processus de génie logiciel. Les motivations de cette recherche ont été longuement expliquées suivies d'une revue de littérature qui a permis de revoir un ensemble d'éléments théoriques indispensables afin d'analyser les données avec plus d'acuité. Une description détaillée du protocole de recherche imaginé et appliqué pour parvenir aux fins de cette étude a ensuite été exposée après quoi, nombreux des résultats obtenus ont été présentés au cours des trois articles constituant le corps du présent document. Chacun des articles permettant de pousser l'analyse du travail collaboratif *ad hoc* un peu plus loin que le précédent, certaines retombées théoriques et pratiques ont été discutées, retombées qui serviront de base à de futures recherches entreprises dans le domaine. Enfin, la dernière portion de ce document a permis de revoir les apports de cette recherche tant au point de vue méthodologique, théorique que pratique afin d'en souligner les contributions significatives à la science du génie logiciel.

Tel que soulevé précédemment, un éventail de questions de recherches futures émergent des analyses présentées au cours de la présente recherche et peuvent être proposées en guise d'ouverture. En voici donc quelques exemples.

- Quelle est la raison pour laquelle une forte proportion de contenus relatifs aux «connaissances» du produit développé et nécessaires à son

développement est observée en dépit des artefacts prescrits par les processus de génie logiciel qui visent à couvrir ce type d'informations?

- Est-t-il plus efficace de conserver une trace du contenu échangé au sein du travail collaboratif *ad hoc*, que ce contenu porte sur le «savoir-faire» ou les «connaissances» nécessaires au développement du produit logiciel, pour y référer ultérieurement ou l'interaction *ad hoc* constitue le mécanisme d'échange le plus efficace?
- Quel type d'artefacts permettrait de colliger et partager efficacement le «savoir-faire» nécessaire pour le développement et la maintenance d'un produit logiciel donné?
- Est-il efficace de prescrire un canal de communication particulier pour un type d'activités collaboratives *ad hoc* particulier tel que certains résultats obtenus dans le cadre de cette recherche laissent à penser?
- Quel est l'impact de la messagerie instantanée sur le travail collaboratif *ad hoc* et les canaux de communication privilégiés?
- Est-il possible de mesurer l'impact réel d'une «interruption» sur le «mental flow» au sens où l'entendent DeMarco et Lister (DeMarco & Lister, 1999);
 - Le temps de récupération de cet état est-il variable d'un concepteur à l'autre?
 - Ce temps de récupération peut-il servir de mesure de «performance» d'un concepteur?

L'étude du travail collaboratif *ad hoc* entreprise dans le cadre des travaux de recherche présentés au cours de ce document aura permis de dresser une première ébauche du modèle théorique du phénomène et ouvre maintenant la porte à de

nombreuses recherches sur le sujet. Comme Fenton et Pfleeger l'indiquaient (Fenton & Pfleeger, 1996) et tel que le rappelait d'Astous (d'Astous, 1999), en tentant de mesurer le «non-mesurable», ce type de recherche ouvre la voie à toute une aire de recherches et discussions qui mènent ultimement à une meilleure compréhension du phénomène étudié.

RÉFÉRENCES

- Adler, R. B., & Towne, N. (1999). Looking out, looking in : interpersonal communication. 9th ed ed.,). Fort Worth ; London : Harcourt Brace.
- Ahern, D. M., Clouse, A., & Turner, R. (2001). CMMI distilled : a practical introduction to integrated process improvement. Boston, [Mass.] ; London : Addison-Wesley.
- Al-Ani, B., & Keith, E. H. (2008). A Comparative Empirical Study of Communication in Distributed and Collocated Development Teams: Global Software Engineering, 2008. ICGSE 2008. IEEE International Conference on. Global Software Engineering, 2008. ICGSE 2008. (pp. 35-44).
- Allen Thomas J. (1977). Managing the flow of technology : technology transfer and the dissemination of technological information within the R&D organization. [Cambridge, Mass. : MIT Press].
- Angrignon, P., & Ruelland G., J. (1995). Civilisation occidentale, histoire et héritages. Montréal: Les Éditions de la Chenelière.
- Babbie, E. R. (1980). Sociology : an introduction. 2d ed ed.,). Belmont, Calif : Wadsworth Pub. Co.
- Babbie, E. R. (2001). The practice of social research (9th ed ed.). Belmont, CA : Wadsworth Thomson Learning.
- Bach, J. (1994). The Immaturity of the CMM. American Programmer, 7(9), 13-18.

- Beck, K. (2000). Extreme programming explained : embrace change. Boston, MA : Addison-Wesley.
- Blake Robert R., & Mouton Jane Srygley. (1964). The managerial grid. Houston ; London : Gulf Pub.
- Brennan, R. L., & Prediger, D. J. (1981). Coefficient Kappa: Some Uses, Misuses, and Alternatives. Educational and Psychological Measurement, 6, 687-699.
- Briand, L., El Eman, K., & Melo, W. L. (1995). (Report No. CS-TR-3498). University of Maryland, College Park, MD20742: Department of Computer Science, University of Maryland.
- Brooks, F. P. Jr. (1987-). No silver bullet; essence and accidents of software engineering. Computer, 20(4), 10-19.
- Brooks, F. P. (1975). The mythical man-month : essays on software engineering. Reprinted with corrections ed.,). Reading, Mass. ; London : Addison-Wesley Pub. Co.
- Carlton, D. W., & Perloff, J. M. (1994). Modern industrial organization. 2nd ed ed.,). New York, NY : HarperCollins College Publishers, c1994.
- Chaplin, C., Goddard, P., MK2 Productions, & Warner Home Video (Firm). (2003). Modern times. [S.l.] ; Burbank, CA : MK2 Editions : Warner Home Video.
- Cherry, S., & Robillard, P. N. (2004a). Communication problems in global software development: spotlight on a new field of investigation. "Third International Workshop on Global Software Development (GSD 2004)" W12S Workshop -

26th International Conference on Software Engineering, 24 May 2004 (pp. 48-52). Edinburgh, Scotland, UK: IEE.

Cherry, S., & Robillard, P. N. (2004b). Empirical Study of Ad Hoc Collaborative Activities in Software Engineering. Proceedings of CSAC 2004 - First International Workshop on Computer Supported Activity Coordination (pp. 116-125). Porto, Portugal: INSTICC Press.

Cherry, S., & Robillard, P.-N. (To be published). Audio video recording of ad hoc interaction from software development team. Proceedings of CHASE 2009 - Workshop on Cooperative and Human Aspects of Software Engineering, May 17 2009 Vancouver, Canada: ACM.

Cherry, S., & Robillard, P. N. (2008). The social side of software engineering--A real ad hoc collaboration network. International Journal of Human-Computer Studies, 66(7), 495-505.

Chong, J., & Siino, R. (2006). Interruptions on software teams: A comparison of paired and solo programmers. Proceedings of the 20th Anniversary ACM Conference on Computer Supported Cooperative Work (pp. 29-38). New York, NY, USA: ACM Press.

Cockburn, A. (2002). Agile software development. Boston, MA : Addison-Wesley.

Cohen, J. (1960). A coefficient of agreement for nominal scales. Educational and Psychological Measurement, 20(1), 37-46.

Cohen, J. (1968). Weighted kappa: Nominal scale agreement with provision for

scaled disagreement or partial agreement. Psychological Bulletin, 70(4), 213-220.

Cragan, J. F., & Wright, D. W. (1986). Communication in small group discussions : an integrated approach. 2nd ed ed.,). St. Paul, MN. : West Publishing.

d'Astous, P., Detienne, F., Visser, W., & Robillard, P. N. (2004-). Changing our view on design evaluation meetings methodology: a study of software technical review meetings. Design Studies, 25(6), 625-55.

d'Astous, P., & Robillard, P. N. (2000). Characterizing implicit information during peer review meetings. Proceedings of the 2000 International Conference on Software Engineering. ICSE 2000 the New Millennium, 4-11 June 2000 (pp. 460-466). Limerick, Ireland: ACM.

d'Astous, P., & Robillard, P. N. (2002a). Empirical study of exchange patterns during software peer review meetings. Information and Software Technology, 44(11), 639-48.

d'Astous, P., & Robillard, P. N. (2002b). Empirical study of exchange patterns during software peer review meetings. Information and Software Technology, 44(11), 639-48.

d'Astous, P., Robillard, P. N., Detienne, F., & Visser, W. (2001). Quantitative measurements of the influence of participant roles during peer review meetings. Empirical Software Engineering, 6(2), 143-159.

d'Astous, P. (1999). Approche de mesure et d'analyse des reunions de revision

technique du processus de genie logiciel. Unpublished doctoral dissertation, Ecole Polytechnique, Montreal (Canada), Canada.

d'Astous, P., & Robillard, P. N. (2000). Characterizing implicit information during Peer Review Meetings. Proceedings - International Conference on Software Engineering: 2000, Jun 4-Jun 11 2000 (pp. 460-466). Limerick, Ireland: Institute of Electrical and Electronics Engineers Computer Society, Los Alamitos, CA, USA.

DeMarco, T. (1982). Controlling software projects : management measurement and estimation. (p. 296). Englewood Cliffs, N.J. : Prentice-Hall.

DeMarco, T. M. a., & Lister, T. R. (1999). Peopleware : productive projects and teams. 2nd ed. ed.,). New York : Dorset House Pub.

Di Penta, M., Harman, M., Antoniol, G., & Qureshi, F. (2007). The Effect of Communication Overhead on Software Maintenance Project Staffing: a Search-Based Approach: Software Maintenance, 2007. ICSM 2007. IEEE International Conference on. Software Maintenance, 2007. ICSM 2007. IEEE International Conference on (pp. 315-324).

Ebert, C., & De Neve, P. (2001). Surviving global software development. Software, IEEE, 18(2), 62-69.

Everett M., R., & Kincaid D., L. (1981). Communication networks : toward a new paradigm for research. New York ; London : Free Press : Collier Macmillan.

Fenton, N. E., & Pfleeger, S. L. (1996). Software metrics : a rigorous and practical

approach (2nd ed ed.). Boston, Mass. :, London : PWS Publishing Company ; International Thomson Computer Press.

Fiedler, F. E. (1967). A theory of leadership effectiveness. New York ; London : McGraw-Hill.

Fischbach, K., Gloor, P.A. & Schoder, D. (2009). Analysis of Informal Communication Networks - A Case Study. Business & Information Systems Engineering. (pp. 140-149). 1(2).

Fisher, C., & Sanderson, P. (1993-). Exploratory sequential data analysis: traditions, techniques and tools. SIGCHI Bulletin, 25(1), 34-40.

Fisher, C., & Sanderson, P. (1996). Exploratory Sequential Data Analysis: Exploring Continuous Observational Data. Interactions, 3(2), 25-34.

Gould, J. (1978). An Experimental Study of Writing, Dictating, and Speaking. Attention and Performance VII . Hillsdale, NJ: Lawrence Earlbaum.

Grinter, R. E., Herbsleb, J. D., & Perry, D. E. (1999). The geography of coordination: dealing with distance in R&D work. Proceedings of GROUP 99: Conference on Supporting Group Work, 14-17 Nov. 1999 (pp. 306-315). Phoenix, AZ, USA: ACM.

Gutwin, C., Penner, R., & Schneider, K. (2004). Group Awareness in Distributed Software Development. Proceedings of the 2004 ACM Conference on Computer Supported Cooperative Work (pp. 72-81). Chicago, Illinois.

Hazzan, O., & Hadar, I. (2008). Why and how can human-related measures support

software development processes? Journal of Systems and Software, 81(7), 1248-1252.

Herbsleb, J. D., & Grinter, R. E. (1999). Splitting the organization and integrating the code: Conway's law revisited. Proceedings of the 21st International Conference on Software Engineering (ICSE '99), 16-22 May 1999 (pp. 85-95). Los Angeles, CA, USA: ACM.

Herbsleb, J. D., & Mockus, A. (2003-). An empirical study of speed and communication in globally distributed software development. IEEE Transactions on Software Engineering, 29(6), 481-494.

Herbsleb, J. D., & Moitra, D. (2001). Global software development. Software, IEEE, 18(2), 16-20.

Hsia, P. (1996). Making software development visible. IEEE Software, 13(3), 23-26.

International Organization for Standardization (ISO). (ISO 9001:2008, Quality management systems - Requirements [Page Web]. URL http://www.iso.org/iso/iso_catalogue/management_standards/iso_9000_iso_14000/iso_9001_2008.htm [26 Avril 2009].

J.S. Karn, & A.J. Cowling. (2005). A study of the effect of disruptions on the performance of software engineering teams. International Symposium on Empirical Software Engineering.

Jett, Q. R., & George, J. M. (2003). *Work interrupted: A closer look at the role of interruptions in organizational life*. Academy of Management Review, 28(3),

494-507.

Jorgensen, D. L. (1989). Participant observation : a methodology for human studies (Applied social research methods series ; No. v. 15). Newbury Park, Calif. ; London : Sage Publications.

Kerbrat-Orecchioni, C. (1998). Les interactions verbales. Paris : Colin.

Kitchenham, B. A., Pfleeger, S. L., Pickard, L. M., Jones, P. W., Hoaglin, D. C., El Emam, K., & Rosenberg, J. (2002). Preliminary guidelines for empirical research in software engineering. IEEE Transactions on Software Engineering, 28(8), 721-734.

Ko, A. J., DeLine, R., & Venolia, G. (2007). Information Needs in Collocated Software Development Teams. Proceedings of the 29th international conference on Software Engineering (pp. 344-353). Washington, DC, USA.

Kobitzsch, W., Rombach, D., & Feldmann, R. L. (2001). Outsourcing in India [software development]. Software, IEEE, 18(2), 78-86.

Kowitz, A. C., & Knutson, T. J. (1980). Decision making in small groups : the search for alternatives. Boston [Mass] ; London [etc.] : Allyn and Bacon.

Krippendorff, K. (2004). Content analysis : an introduction to its methodology. 2nd ed ed.,). Thousand Oaks, Calif. ; London : Sage.

Kroll, B. M. (1978). Cognitive Egocentrism and the Problem of Audience Awareness in Written Discourse. Research in the Teaching of English, 12(3), 269-281.

Kruchten, P. (2004). Analyzing intercultural factors affecting global software

development - a position paper. "Third International Workshop on Global Software Development (GSD 2004)" W12S Workshop - 26th International Conference on Software Engineering, 24 May 2004 (pp. 59-62). Edinburgh, Scotland, UK: IEE.

Kruchten, P. (2000). The rational unified process : an introduction (2nd ed ed.). (Addison-Wesley object technology series . Boston, MA : Addison-Wesley.

Larman, C. (2002). Applying UML and patterns : an introduction to object-oriented analysis and design and the unified process (2nd ed ed.). Upper Saddle River, NJ : Prentice Hall PTR.

LaToza, T. D., Venolia, G., & DeLine, R. (2006). Maintaining Mental Models: A Study of Developer Work Habits. Proceedings of the 28th international conference on Software engineering (pp. 492-501). Shanghai, China.

Lofland, J., & Lofland, L. H. (1995). Analyzing social settings : a guide to qualitative observation and analysis. 3rd ed ed.,). Belmont, Calif. ; London : Wadsworth Publishing.

Melnik, G., & Maurer, F. (2004). Direct verbal communication as a catalyst of agile knowledge sharing. Proceedings of the Agile Development Conference. ADC 2004, 22-26 June 2004 (pp. 21-31). Salt Lake City, UT, USA: IEEE Comput. Soc.

Mockus, A., & Herbsleb, J. D. (2002). Expertise Browser: a quantitative approach to identifying expertise. Proceedings of the 24th International Conference on Software Engineering. ICSE 2002, 19-25 May 2002 (pp. 503-512). Orlando,

FL, USA: ACM.

Naur, P., & Randell, B. (1969). Software engineering: report on a conference sponsored by the NATO Science committee, Garmish, Germany 7-11 October 19. NATO.

NSERC, N. S. a. E. R. C. o. C. (2005) Tri-Council Policy Statement: Ethical Conduct for Research Involving Humans. In. *Interagency Advisory Panel on Research Ethics* [Page Web]. URL <http://www.pre.ethics.gc.ca/english/policystatement/policystatement.cfm> [15 février 2009].

Paulk, M. C., & Carnegie Mellon University. Software Engineering Institute. (1994). The capability maturity model : guidelines for improving the software process. (p. 441 p.). Reading, Mass. ; Wokingham, England : Addison-Wesley Pub. Co.

Perreault, W. D. Jr., & Leigh Laurence E. (1989). Reliability of Nominal Data Based on Qualitative Judgements. Journal of Marketing Research, 26, 135-148.

Perry, D. E., Staudenmayer, N. A., & Votta, L. G. (1994-). People, organizations, and process improvement. IEEE Software, 11(4), 36-45.

Popping, R. (1988). On agreement indices for nominal data. W. E. Saris, I. N. Gallhofer, & International Sociological Association. Sociometric research (Vol. Data collection and scalingpp. 90-105). Basingstoke : Macmillan.

Potts, C., & Catledge, L. (1996). Collaborative conceptual design: a large software

project case study. Computer Supported Cooperative Work (CSCW), 5(4), 415-45.

Pressman, R. S. (2001). Software engineering : a practitioner's approach (5th ed ed.). (McGraw-Hill series in computer science . Boston, Mass. : McGraw Hill.

Project Management Institute. (2000). A guide to the project management body of knowledge (PMBOK guide). 2000 ed ed.,). Newtown Square, Penn., USA : Project Management Institute.

Ritchie, D. (1986). The computer pioneers : the making of the modern computer. New York : Simon and Schuster.

Robillard, P. N. (2005). Opportunistic Problem Solving in Software Engineering. Software, IEEE, 22(6), 60-67.

Robillard, P. N., d'Astous, P., Detienne, F., & Visser, W. (1998). Measuring cognitive activities in software engineering. Proceedings of the 1998 International Conference on Software Engineering. Forging New Links, 19-25 April 1998 (pp. 292-300). Kyoto, Japan: IEEE Comput. Soc.

Robillard, P. N., & Robillard, M. P. (2000). Types of collaborative work in software engineering. Journal of Systems and Software, 53(3), 219-224.

Robillard, P. N. (1996). (Report No. EPM/RT-96/18). Montréal: École Polytechnique de Montréal.

Robillard, P. N., Kruchten, P., & d'Astous, P. (2003). Software engineering process with the UPEDU. Boston : Addison Wesley.

- Sanderson, P. M., & Fisher, C. (1994). Exploratory sequential data analysis: foundations. Human-Computer Interaction, 9(3-4), 251-317.
- Sarma, A., Maccherone, L., Wagstrom, P., & Herbsleb, J. (2009). Tesseract: interactive visual exploration of socio-technical relationships in software development. 2009 31st International Conference on Software Engineering (ICSE 2009), 16-24 May 2009 (pp. 23-33). Piscataway, NJ, USA: IEEE.
- Seaman, C. B. (1999-). Qualitative methods in empirical studies of software engineering. IEEE Transactions on Software Engineering, 25(4), 557-572.
- Seaman, C. B., & Basili, V. R. (1997). Communication and organization in software development: an empirical study. IBM Systems Journal, 36(4), 550-563.
- Seaman, C. B., & Basili, V. R. (1998). Communication and Organization: An Empirical Study of Discussion in Inspection Meetings. IEEE Transactions on Software Engineering, 7(24), 559-572.
- Shaw, M. E. (1981). Group dynamics : the psychology of small group behaviour. 3rd ed ed.,). New York ; London : McGraw-Hill.
- Singer, J., & Vinson, N. G. (2002). Ethical Issues in Empirical Studies of Software Engineering. IEEE Transactions on Software Engineering , 28(12), 1171-1180.
- Steinzor, B. (1950). The spatial factor in face to face discussion groups. J Abnorm Soc Psychol, 45(3), 552-5.
- Tichy, W. F. (1998). Should computer scientists experiment more? Computer,

31(5), 32-40.

Treude, C., & Storey, M.-A. (2009). How tagging helps bridge the gap between social and technical aspects in software development. 2009 31st International Conference on Software Engineering (ICSE 2009), 16-24 May 2009 (pp. 12-22). Piscataway, NJ, USA: IEEE.

Tuckman, B. W. (1965). Developmental Sequence in Small Groups. Psychol Bull, 63, 384-399.

Walker, R. J., Briand, L. C., Notkin, D., Seaman, C. B., & Tichy, W. F. (2003). Panel: Empirical validation - What, why, when, and how. Proceedings - International Conference on Software Engineering, May 3-10 2003 (pp. 721-722). Portland, OR, United States: Institute of Electrical and Electronics Engineers Computer Society.

Watzlawick, P., Bavelas, J. B., & Jackson, D. D. (1968). Pragmatics of human communication : a study of interactional patterns, pathologies, and paradoxes. London : Faber.

Wayt Gibbs, W. (1994). Software's chronic crisis. Scientific American (International Edition), 271(3), 72-81.

World Wide Web Consortium (W3C). (Extensible Markup Language (XML) [Page Web]. URL <http://www.w3.org/XML/> [5 octobre, 2005a].

World Wide Web Consortium (W3C). (XML Schema Part 0: Primer Second Edition [Page Web]. URL <http://www.w3.org/TR/xmlschema-0/> [9 octobre, 2005b].

Annexe A

**Certificat d'acceptation du projet de recherche par le
Comité d'éthique de la recherche avec des sujets humains
de l'École Polytechnique de Montréal**



**Comité d'éthique de la
recherche avec des
sujets humains**

Adresse civique
Campus de
l'Université de Montréal
2900, boul. Edouard-Montpetit
École Polytechnique
2500, chemin de Polytechnique
Montréal (Québec) H3T 1J4

Adresse postale
C.P. 6079, succursale Centre-ville
Montréal (Québec) H3C 3A7

Téléphone : (514) 340-4852
Télécopieur : (514) 340-4611

École affiliée à
l'Université de Montréal

Membres du comité :

Mme Ginette Denicourt, IRSST
M. Daniel Imbeau, MAGI
M. Bernard Lapierre, MAGI
Dr. André Phaneuf, Fac. Méd. Dent.
Pierre Savard, IGB, président

**CERTIFICAT D'ACCEPTATION D'UN PROJET DE
RECHERCHE PAR LE
COMITÉ D'ÉTHIQUE DE LA RECHERCHE AVEC
DES SUJETS HUMAINS DE L'ÉCOLE POLYTECHNIQUE**

Montréal, le 19 août 2005

M. Sébastien CHERRY
Département de génie informatique
École Polytechnique de Montréal

N/Réf : Dossier CÉR-05/06-05

Cher M. Cherry,

J'ai le plaisir de vous informer que le *Comité d'éthique de la recherche avec des sujets humains de l'École Polytechnique (CÉRSHÉP)* a approuvé, lors de sa réunion du 19 août 2005, votre projet de recherche intitulé :

*« Mesure et analyse des activités collaboratrices ad hoc
retrouvées au cours d'un processus de génie logiciel »*

Il est entendu que le présent certificat est valable pour le projet tel que soumis au CÉRSHÉP. Le CÉRSHÉP devra être informé de toute modification qui pourrait être apportée ultérieurement au protocole expérimental, de même que de tout problème imprévu pouvant avoir une incidence sur la santé, le bien-être et la sécurité des personnes impliquées dans le projet de recherche (sujets, professionnels de recherche ou chercheurs).

Nous vous prions également de nous faire parvenir un bref rapport annuel ainsi qu'un avis à la fin de vos travaux.

Je vous souhaite bonne chance dans vos travaux de recherche,

Pierre Savard
Président

Comité d'éthique de la recherche avec des sujets humains

C.C. : M. Jean Choquette, Bernard Lapierre, Pierre N. Robillard

Annexe B

**Plan de l'espace de travail de l'environnement observé et
emplacement de la caméra et des microphones**

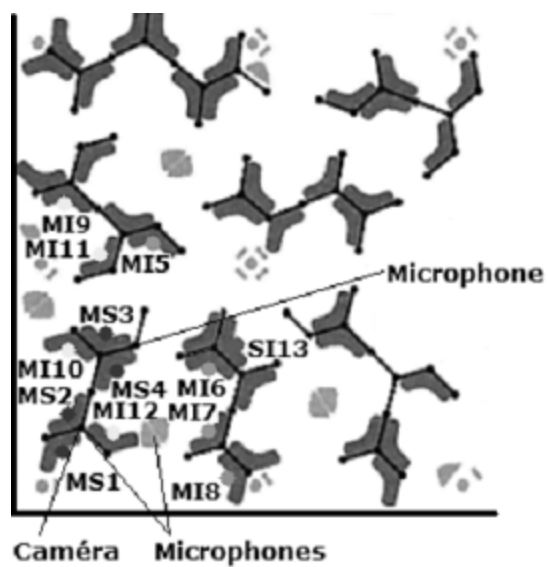


Figure B.1 Plan de l'espace de travail de l'environnement observé et emplacement de la caméra et des microphones

Annexe C

Liste des enregistrements audio-vidéos et échantillonnages réalisés pour l'analyse

Tableau C.1 Liste des enregistrements audio-vidéos exclus pour accoutumer les sujets à la présence de la caméra

Date	Durée
Jeudi, 11 septembre	0:49:59
Vendredi, 12 septembre	3:00:00
	3:00:00
Lundi, 15 septembre	3:00:00
	3:00:00
Mardi, 16 septembre	3:00:00
	3:00:00
Mercredi, 17 septembre	3:00:00
	1:22:32
Jeudi, 18 septembre	3:00:00
	2:45:15
Vendredi, 19 septembre	3:00:00
	0:30:00
	1:50:05
Lundi, 22 septembre	2:00:00
	3:00:00
Mardi, 23 septembre	3:00:00
	3:00:00

Mercredi, 24 septembre	3:00:00
	3:00:00
Jeudi, 25 septembre	2:45:00
	2:45:00
Vendredi, 26 septembre	3:00:00
	3:00:00
Lundi, 29 septembre	3:00:00
	3:00:00
Mardi, 30 septembre	3:00:00
Mercredi, 1 octobre	3:00:00
	2:45:00

Tableau C.2 Liste des enregistrements audio-vidéos utilisés pour fins d'analyse

Date	Durée	Utilisé pour :			
		Analyse	Validation intra-codeur 1	Validation intra-codeur 2	Validation inter-codeurs
Jeudi, 2 octobre	2:30:00	X	X	X	X
	2:30:00				
Vendredi, 3 octobre	2:45:00	X			
	2:45:00	X	X	X	X
Lundi, 6 octobre	2:45:00	X			
	2:00:00				
Mardi, 7 octobre	2:30:00				
	2:50:00				
Mercredi, 8 octobre	3:00:00				
Jeudi, 9 octobre	3:00:00				
	1:45:00				
Vendredi, 10 octobre	3:00:00	X		X	X
	3:00:00	X			
Mardi, 14 octobre	3:00:00	X			
	3:00:00	X		X	
Mercredi, 15 octobre	3:00:00				
	2:30:00				
Jeudi, 16 octobre	2:40:00				

	2:10:00				
Vendredi, 17 octobre	2:40:00				
Lundi, 20 octobre	3:00:00				
	3:10:00				
Mardi, 21 octobre	3:00:00				
	3:00:00				
Mercredi, 22 octobre	2:00:00				
	2:00:00				
Jeudi, 23 octobre	3:00:00	X			
	3:00:00				
Vendredi, 24 octobre	3:00:00				
	3:00:00	X			
Lundi, 27 octobre	3:00:00				
	3:00:00				
Mardi, 28 octobre	3:00:00	X		X	
	2:00:00				
Mercredi, 29 octobre	2:00:00				
	2:14:44				
Jeudi, 30 octobre	3:00:00				
	2:40:00	X			
Vendredi, 31 octobre	3:00:00				
	2:00:00				
Lundi, 3 novembre	3:00:00	X			

3:00:00

Les rangées grisées représentent les enregistrements exclus en raison d'absences de participants.

Annexe D

**Images écran de l'outil conçu pour l'analyse exploratoire
et séquentielle des données**

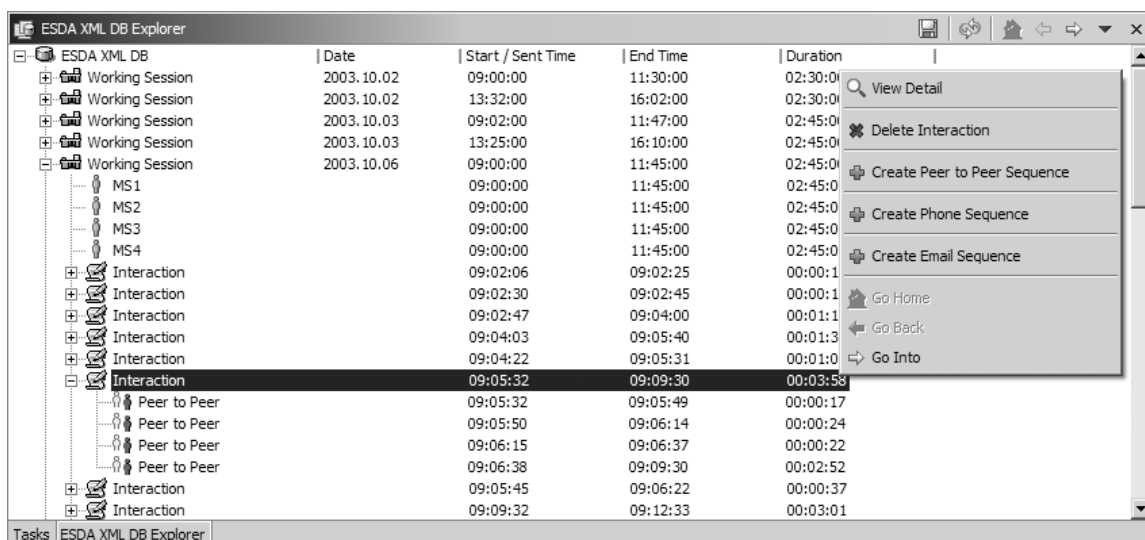


Figure D.1 Outil d'analyse exploratoire et séquentielle de données
– outil de saisie de données

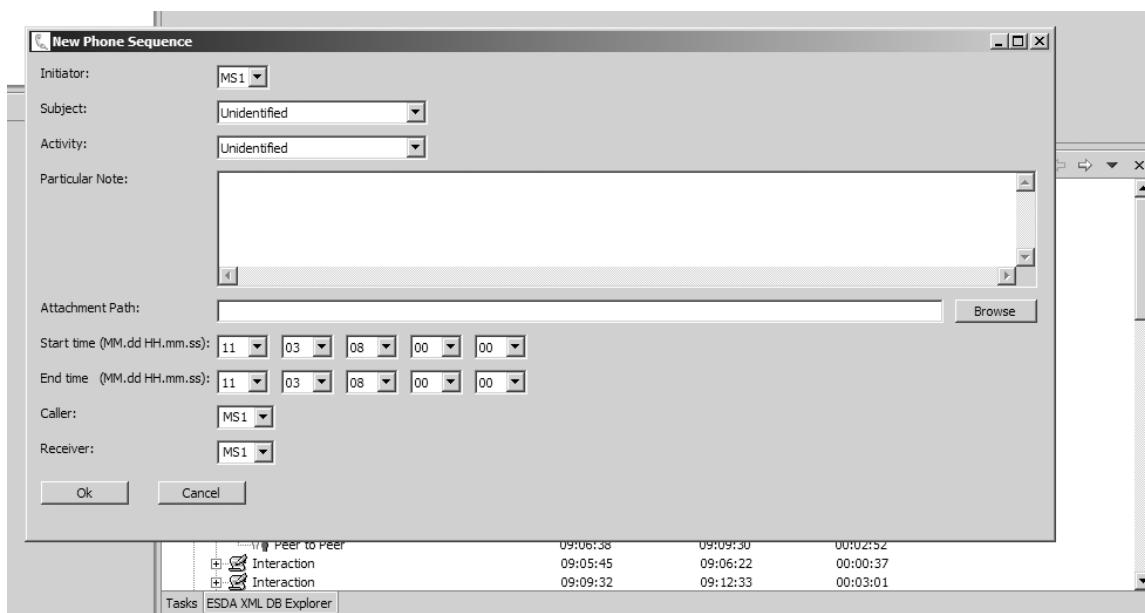


Figure D.2 Outil d'analyse exploratoire et séquentielle de données
– outil de saisie de données et écran de saisie

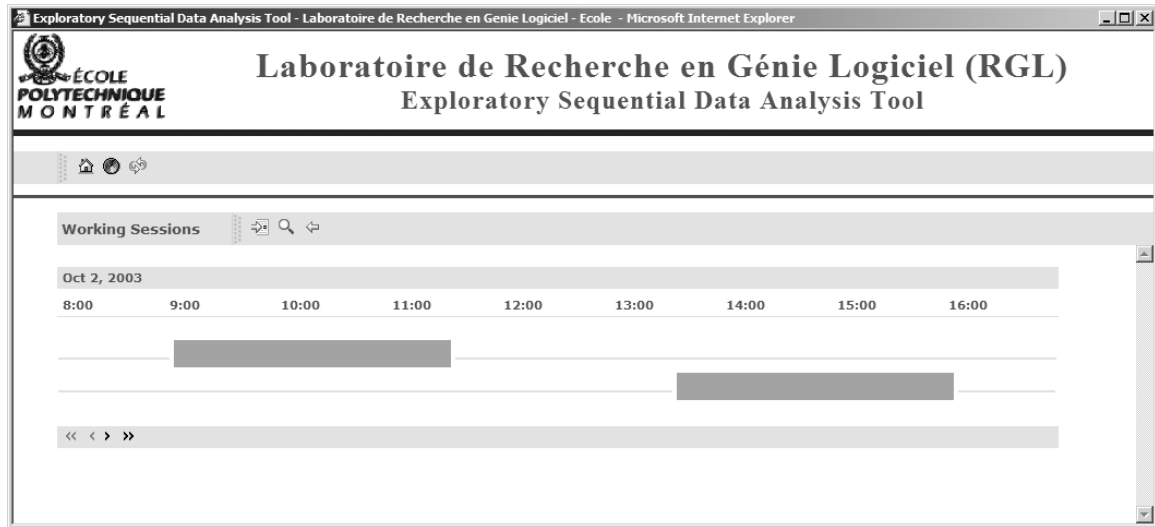


Figure D.3 Outil d'analyse exploratoire et séquentielle de données – diagramme gant des sessions de travail analysées

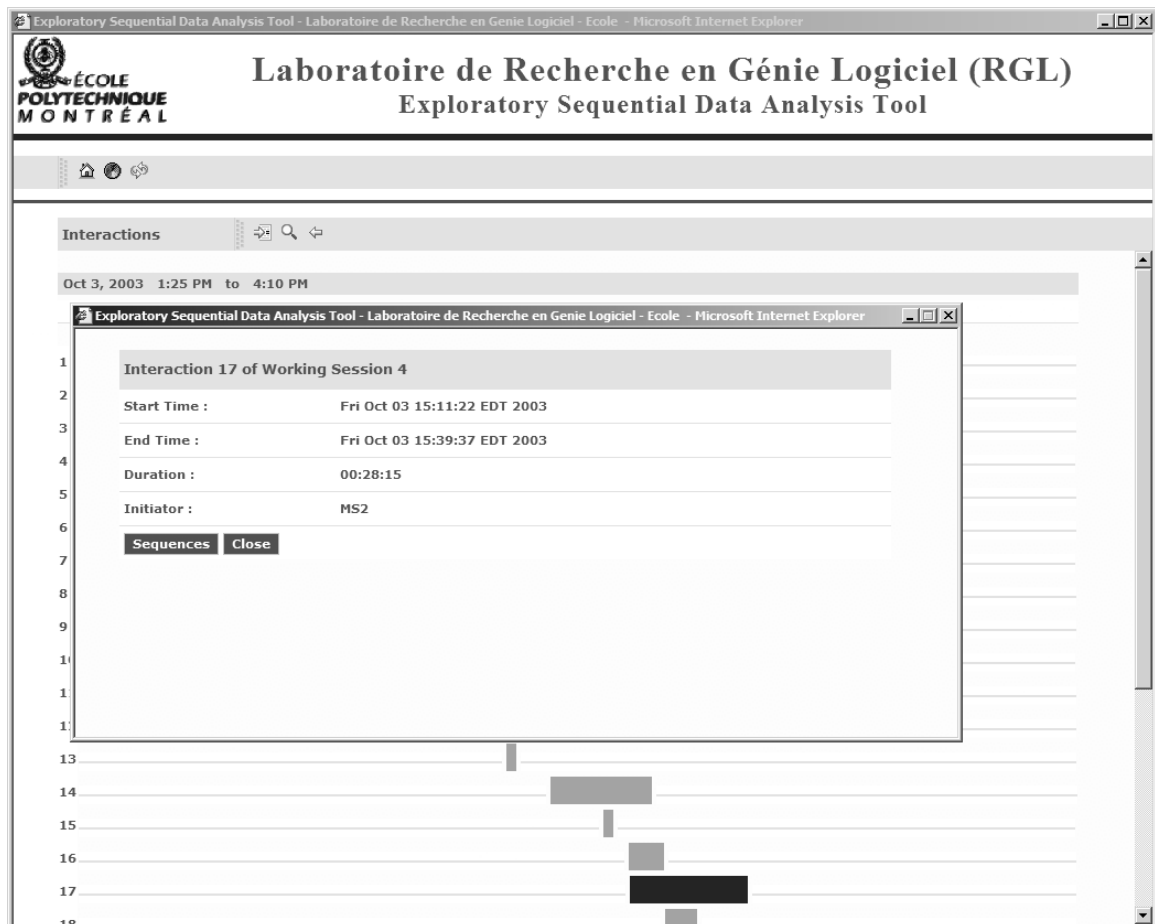


Figure D.4 Outil d'analyse exploratoire et séquentielle de données – diagramme gant des interactions observées lors d'une session de travail analysée et détails d'une interaction

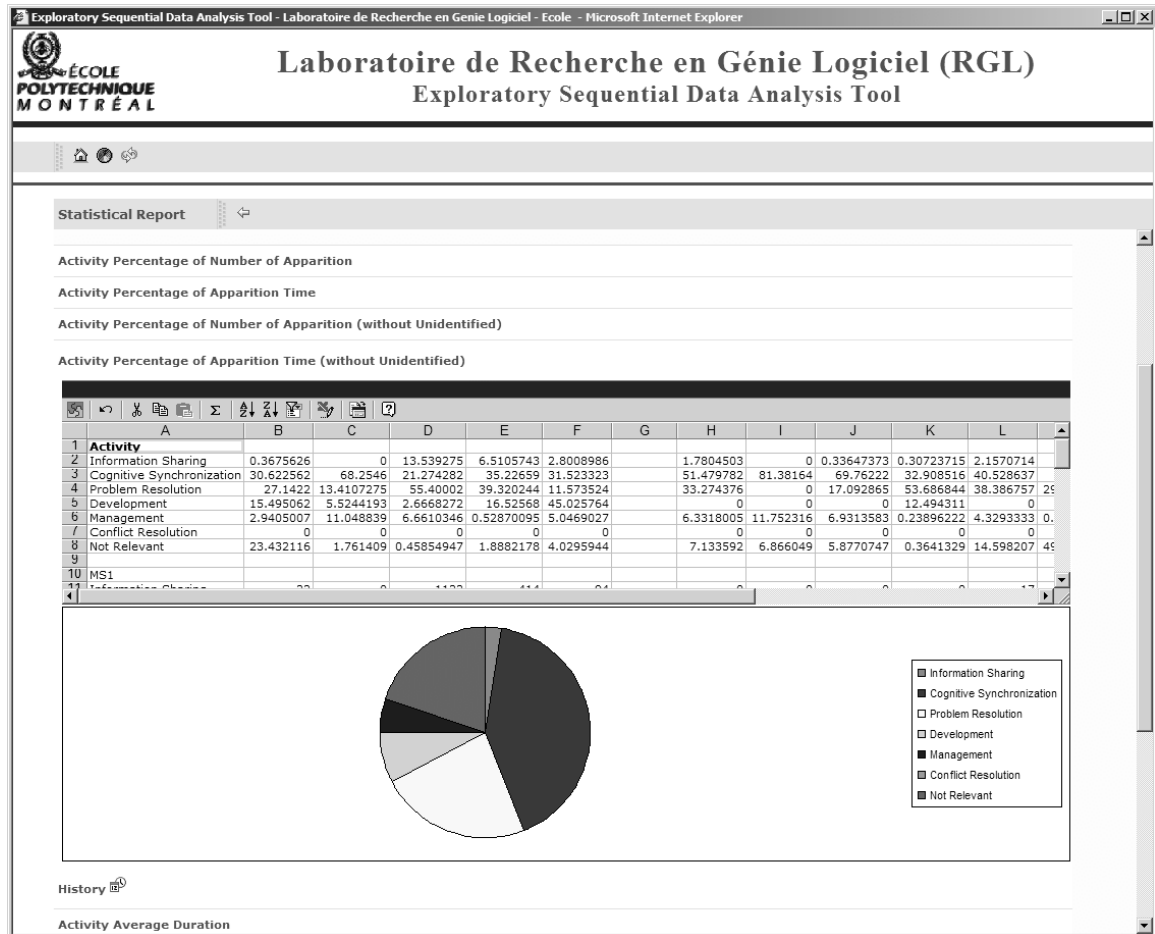


Figure D.5 Outil d'analyse exploratoire et séquentielle de données
– rapport d'analyse des sessions de travail analysées

Annexe E

Échantillon type de données utilisées pour l'analyse

Échantillon 2 Exemple type étendu de données utilisées pour l'analyse

```
<working-session>
  <start-time
    year="2003" month="9" day="2"
    hour="9" minute="0" second="0"/>
  <end-time
    year="2003" month="9" day="2"
    hour="11" minute="30" second="0"/>
  <participant name="MS1">
    <start-time
      year="2003" month="9" day="2"
      hour="9" minute="0" second="0"/>
    <end-time
      year="2003" month="9" day="2"
      hour="11" minute="30" second="0"/>
  </participant>
  <participant name="MS2">
    <start-time
      year="2003" month="9" day="2"
      hour="9" minute="0" second="0"/>
    <end-time
      year="2003" month="9" day="2"
      hour="11" minute="30" second="0"/>
  </participant>
  <participant name="MS3">
    <start-time
      year="2003" month="9" day="2"
```

```

        hour="9" minute="3" second="0"/>
    <end-time
        year="2003" month="9" day="2"
        hour="11" minute="30" second="0"/>
</participant>
<participant name="MS4">
    <start-time
        year="2003" month="9" day="2"
        hour="9" minute="0" second="0"/>
    <end-time
        year="2003" month="9" day="2"
        hour="11" minute="30" second="0"/>
</participant>
<interaction-list>
    <interaction>
        <sequence-list>
            <face2face-sequence initiator="?">
                <start-time
                    year="2003" month="9" day="2"
                    hour="9" minute="0" second="0"/>
                <end-time
                    year="2003" month="9" day="2"
                    hour="9" minute="3" second="0"/>
                <stakeholders list="MS4 MI7"/>
                <movement place="MS4"/>
                <purpose subject="Specific Topic 3"
                    activity="Cognitive Synchronization"/>
                <particular-note>...</particular-note>

```

```

        </face2face-sequence>
    </sequence-list>
</interaction>
<interaction>
    <sequence-list>
        <face2face-sequence initiator="MS3">
            <start-time year="2003" month="9" day="2"
                hour="9" minute="4" second="0"/>
            <end-time year="2003" month="9" day="2"
                hour="9" minute="4" second="41"/>
            <stakeholders list="MS3 MS4"/>
            <movement place="MS3"/>
            <purpose subject="Specific Topic 6"
                activity="Cognitive Synchronization"/>
            <particular-note>...</particular-note>
        </face2face-sequence>
        <face2face-sequence initiator="MS4">
            <start-time year="2003" month="9" day="2"
                hour="9" minute="4" second="42"/>
            <end-time year="2003" month="9" day="2"
                hour="9" minute="5" second="58"/>
            <stakeholders list="MS3 MS4"/>
            <movement place="MS3"/>
            <purpose subject="Specific Topic 6"
                activity="Problem Resolution"/>
            <particular-note>...</particular-note>
        </face2face-sequence>
        <face2face-sequence initiator="MI7">

```

```

        <start-time year="2003" month="9" day="2"
            hour="9" minute="5" second="59"/>
        <end-time year="2003" month="9" day="2"
            hour="9" minute="8" second="15"/>
        <stakeholders list="MS3 MS4 MI7"/>
        <movement place="MS3"/>
        <purpose subject="Configuration Management"
            activity="Cognitive Synchronization"/>
        <particular-note>...</particular-note>
    </face2face-sequence>
    <face2face-sequence initiator="?">
        <start-time year="2003" month="9" day="2"
            hour="9" minute="8" second="16"/>
        <end-time year="2003" month="9" day="2"
            hour="9" minute="9" second="3"/>
        <stakeholders list="MS3 MS4"/>
        <movement place="MS3"/>
        <purpose subject="Configuration Management"
            activity="Cognitive Synchronization"/>
        <particular-note>...</particular-note>
    </face2face-sequence>
    ...
</sequence-list>
</interaction>
...
<interaction>
    <sequence-list>
        <email-sequence>

```

```

    <sent-time year="2003" month="9" day="2"
        hour="9" minute="6" second="0"/>
    <stakeholders sender="AI22"
        receivers="MS2 AI23"/>
    <purpose subject="Configuration Management"
        activity="Cognitive Synchronization"
        message-subject="..."/>
    <attachment path="..."/>
    <written-size nbwords="19"/>
    <to-read-size nbwords="31"/>
    <index value="4"/>
    </email-sequence>
</sequence-list>
</interaction>
...
</interaction-list>
</working-session>

```

Le sujet des courriels et les commentaires ont été retirés pour des raisons de confidentialité.

Annexe F

**Schéma XML utilisé pour la sérialisation et validation des
données encodées sur support physique**

```

<?xml version="1.0"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:element name="working-session-list">
    <xs:complexType>
      <xs:sequence>
        <xs:element ref="working-session" minOccurs="1"
          maxOccurs="unbounded"/>
      </xs:sequence>
    </xs:complexType>
  </xs:element>

  <xs:element name="working-session">
    <xs:complexType>
      <xs:sequence>
        <xs:element name="start-time" type="timeType"/>
        <xs:element name="end-time" type="timeType"/>
        <xs:element name="participant" minOccurs="1"
          maxOccurs="unbounded">
          <xs:complexType>
            <xs:sequence>
              <xs:element name="start-time"
                type="timeType"/>
              <xs:element name="end-time"
                type="timeType"/>
            </xs:sequence>
            <xs:attribute name="name"
              type="xs:string"/>
          </xs:complexType>
        </xs:element>
        <xs:element ref="interaction-list" minOccurs="0"/>
      </xs:sequence>
    </xs:complexType>
  </xs:element>

  <xs:element name="interaction-list">
    <xs:complexType>
      <xs:sequence>

```

```

        <xs:element ref="interaction" minOccurs="1"
            maxOccurs="unbounded"/>
    </xs:sequence>
</xs:complexType>
</xs:element>

<xs:element name="interaction">
    <xs:complexType>
        <xs:sequence>
            <xs:element ref="sequence-list" minOccurs="1"/>
        </xs:sequence>
    </xs:complexType>
</xs:element>

<xs:element name="sequence-list">
    <xs:complexType mixed="true">
        <xs:sequence>
            <xs:any minOccurs="1" maxOccurs="unbounded"/>
        </xs:sequence>
    </xs:complexType>
</xs:element>

<xs:element name="face2face-sequence">
    <xs:complexType>
        <xs:sequence>
            <xs:element name="start-time" type="timeType"/>
            <xs:element name="end-time" type="timeType"/>
            <xs:element name="stakeholders">
                <xs:complexType>
                    <xs:attribute name="list"
                        type="xs:string"/>
                </xs:complexType>
            </xs:element>
            <xs:element name="movement" minOccurs="0">
                <xs:complexType>

```



```

        <xs:attribute name="place"
            type="xs:string"/>
    </xs:complexType>
</xs:element>
<xs:element name="purpose">
    <xs:complexType>
        <xs:attribute name="subject"
            type="xs:string"/>
        <xs:attribute name="activity"
            type="xs:string"/>
    </xs:complexType>
</xs:element>
<xs:element name="particular-note" type="xs:string"
    minOccurs="0"/>
<xs:element name="attachment" minOccurs="0">
    <xs:complexType>
        <xs:attribute name="path"
            type="xs:anyURI"/>
    </xs:complexType>
</xs:element>
</xs:sequence>
<xs:attribute name="initiator" type="xs:string"/>
</xs:complexType>
</xs:element>

<xs:element name="phone-sequence">
    <xs:complexType>
        <xs:sequence>
            <xs:element name="start-time" type="timeType"/>
            <xs:element name="end-time" type="timeType"/>
            <xs:element name="stakeholders">
                <xs:complexType>
                    <xs:attribute name="caller"
                        type="xs:string"/>
                    <xs:attribute name="receiver"
                        type="xs:string"/>
                </xs:complexType>
            </xs:element>
        </xs:sequence>
    </xs:complexType>
</xs:element>

```

```

</xs:element>
<xs:element name="purpose">
  <xs:complexType>
    <xs:attribute name="subject"
      type="xs:string"/>
    <xs:attribute name="activity"
      type="xs:string"/>
  </xs:complexType>
</xs:element>
<xs:element name="particular-note" type="xs:string"
  minOccurs="0"/>
<xs:element name="attachment" minOccurs="0">
  <xs:complexType>
    <xs:attribute name="path"
      type="xs:anyURI"/>
  </xs:complexType>
</xs:element>
</xs:sequence>
<xs:attribute name="initiator" type="xs:string"/>
</xs:complexType>
</xs:element>

<xs:element name="email-sequence">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="sent-time" type="timeType"/>
      <xs:element name="stakeholders">
        <xs:complexType>
          <xs:attribute name="sender"
            type="xs:string"/>
          <xs:attribute name="receivers"
            type="xs:string"/>
        </xs:complexType>
      </xs:element>
      <xs:element name="purpose">
        <xs:complexType>
          <xs:attribute name="subject"

```

```

        type="xs:string"/>
        <xs:attribute name="activity"
            type="xs:string"/>
        <xs:attribute name="message-subject"
            type="xs:string"/>
    </xs:complexType>
</xs:element>
<xs:element name="particular-note" type="xs:string"
    minOccurs="0"/>
<xs:element name="attachment" minOccurs="0">
    <xs:complexType>
        <xs:attribute name="path"
            type="xs:anyURI"/>
    </xs:complexType>
</xs:element>
<xs:element name="written-size">
    <xs:complexType>
        <xs:attribute name="nbwords"
            type="xs:nonNegativeInteger"/>
    </xs:complexType>
</xs:element>
<xs:element name="to-read-size">
    <xs:complexType>
        <xs:attribute name="nbwords"
            type="xs:nonNegativeInteger"/>
    </xs:complexType>
</xs:element>
<xs:element name="index">
    <xs:complexType>
        <xs:attribute name="value"
            type="xs:positiveInteger"/>
    </xs:complexType>
</xs:element>
</xs:sequence>
    <xs:attribute name="initiator" type="xs:string"/>
</xs:complexType>
</xs:element>

```

```

<xs:complexType name="timeType">
  <xs:attribute name="year">
    <xs:simpleType>
      <xs:restriction base="xs:integer">
        <xs:minInclusive value="2003"/>
        <xs:maxInclusive value="2003"/>
      </xs:restriction>
    </xs:simpleType>
  </xs:attribute>
  <xs:attribute name="month">
    <xs:simpleType>
      <xs:restriction base="xs:integer">
        <xs:minInclusive value="1"/>
        <xs:maxInclusive value="12"/>
      </xs:restriction>
    </xs:simpleType>
  </xs:attribute>
  <xs:attribute name="day">
    <xs:simpleType>
      <xs:restriction base="xs:integer">
        <xs:minInclusive value="1"/>
        <xs:maxInclusive value="31"/>
      </xs:restriction>
    </xs:simpleType>
  </xs:attribute>
  <xs:attribute name="hour">
    <xs:simpleType>
      <xs:restriction base="xs:integer">
        <xs:minInclusive value="0"/>
        <xs:maxInclusive value="23"/>
      </xs:restriction>
    </xs:simpleType>
  </xs:attribute>
  <xs:attribute name="minute">
    <xs:simpleType>
      <xs:restriction base="xs:integer">

```

```
        <xs:minInclusive value="0"/>
        <xs:maxInclusive value="59"/>
    </xs:restriction>
</xs:simpleType>
</xs:attribute>
<xs:attribute name="second">
    <xs:simpleType>
        <xs:restriction base="xs:integer">
            <xs:minInclusive value="0"/>
            <xs:maxInclusive value="59"/>
        </xs:restriction>
    </xs:simpleType>
</xs:attribute>
</xs:complexType>
</xs:schema>
```

Annexe G

Audio Video Recording of *Ad hoc* Software Development

Team Interactions

**Proceedings of CHASE 2009 - International Workshop on
Cooperative and Human Aspects of Software Engineering,
31st International Conference on Software Engineering**

Vancouver, Canada, 2009

Audio-Video Recording of *ad hoc* Software Development Team Interactions

Sébastien Cherry, Pierre N. Robillard

Department of Computer and Software Engineering

École Polytechnique de Montréal,

C.P. 6079, Succ. Centre-Ville, Montréal, Qc, Canada, H3C 3A7

{Sebastien.Cherry, Pierre-N.Robillard} @polymtl.ca

Abstract

Ad hoc interactions characterize the natural behaviors observed in any teamwork situation. Our objective is to find out how these interactions are related to the roles team mates play. Observations on the phenomenon are based on audio-video recordings of software development team interactions occurring in a large software development organization with a highly standardized software development process in place. Detailed observations were recorded on the activities of four of the twelve members of the development team. Thirty-five hours of audio-video recordings were analyzed, in which a total of 404 face-to-face ad hoc collaborative interactions were observed. The results provided a quantitative demonstration of the impact of a newcomer, a “guru”, and a project manager on the team dynamics. This study, although conducted in a specific professional environment, can provide useful information for a better understanding of the face-to-face communication needs of a software development team.

1. Introduction

Ad hoc collaboration is a natural human behavior observed in any teamwork situation. Some authors consider it an intrusion. Jett *et al.* [1] define an intrusion as “an unexpected encounter initiated by another person that interrupts the flow and continuity of an individual’s work and brings that work to a temporary halt.” Intrusions tend to be immediate in nature, and much of their disruptive impact derives from a perceived need to respond promptly to the needs of the intruder, in this case the interrupter.

The overall objective of this study is to better understand *ad hoc* interruptions. Our first specific objective is to quantify the interruption phenomenon and to find out how *ad hoc* interactions are related to the roles team mates play.

Perry *et al.* have tried to describe how developers spend their time as a software development project unfolds [2]. Their studies show, among other things, that developers spend 75 minutes per day on unplanned informal communications. This observation was later supported by Robillard and Robillard in their study on the different types of collaborative work in software engineering [3], which they identified as mandatory meetings, called or scheduled meetings, *ad hoc* meetings, and individual work. A later study by D’Astous *et al.* [4] analyzed the patterns of exchange between team mates during called meetings and during meetings scheduled in advance, such as peer-review meetings. For this, they built a model to represent the qualitative and quantitative importance of various exchanges occurring during these meetings. In contrast, the study presented in this paper analyzes *ad hoc* interactions occurring during individual work, which can also be referred to as interruptions.

Ad hoc interactions have been observed to account for more than 40% of the time spent on a software project [2, 4]. The importance of *ad hoc* activities to team dynamics is shared by many authors and studied from various perspectives [5, 6]. Despite this fairly large consensus, no known study has yet described and characterized *ad hoc* collaborative activities and their communication patterns. This is what we have endeavored to do in the research described in this paper.

Many aspects of collaborative work have already been studied. Seaman *et al.*, and more recently Gutwin *et al.* [7, 8], have demonstrated that co-location, mobility, and the opportunity to communicate

spontaneously are major factors in determining the quality of awareness information and facilitating the solution of coordination problems.

Communication patterns are sometimes polarized in favor of those individuals who are likely to possess legacy knowledge about a project that is already under way. Such “gurus” may serve as a magnet for *ad hoc* interaction from team members digging for specific information [9].

Previous research has documented the fragmented nature of software development work, with frequent interruptions and instances of coordination [10]. Ko *et al.* [11] analyzed detailed activity logs associated with the search for information on the part of developers. They consider face-to-face *ad hoc* activities and e-mail as causes of task switching.

Taken together, these studies suggest that *ad hoc* interruptions are a natural and necessary component of collaborative knowledge work. This study considers external interruptions, which involve interaction with other individuals. These interruptions may involve movement, where team members physically visit one another’s cubicles, or verbal communication, where individuals interact verbally while they are still seated at their workstations.

2. Methodology

This research was conducted in exploratory mode. Sanderson *et al.* have described three approaches to the study of human behavior in the methodological tradition [12]. The *behavioral* approach is concerned with human behavior in its natural environment. The *cognitive* approach focuses on human cognitive performance in the accomplishment of particular tasks. The *social* approach focuses on the interactions and communications of individuals in their environment, i.e. human-human or human-computer. This approach favors field observations and values direct observer participation in an effort to cover representative situations, instead of relying on a statistical sampling.

Our methodology has been inspired by the social approach. Our subjects were sampled using a judged sampling method, so that we would study subjects who were more likely to provide us with relevant material to analyze.

2.1. Observer or participant

The participant-observation method raises an important question, which is the extent to which the observer himself participates in the group of

individuals being observed. In fact, we are dealing with a continuum which stretches from the complete participant, who takes part in all the activities of the group wholeheartedly, and even becomes a fully fledged member of that group, to the complete observer, who does not take part in the activities of the group at all and remains on the fringe. This raises the issue of objectivity vs. subjectivity in qualitative research; however, the dichotomy lost some of its relevance when a consensus emerged among participant-observers that, in order to avoid misunderstanding and misinterpretation of the observations, the observer must grow closer, both physically and socially, to those being observed, who are then called “insiders” [13]. Most of the previous studies in the field of collaborative interactions are based on the observer approach.

One of the main advantages of the participant-observation method is that it can generate highly valid results, since the definition of the underlying concepts is inspired directly from the reality of those being observed (whom we refer to as *observees*). However, the reliability of participant-observation research suffers somewhat, because the collection methods used are usually adapted to the field context, which makes reproducibility more difficult. Nevertheless, reliability in participant-observation research can also be enhanced by the solidity and veracity of the resulting discoveries [14].

During the course of this research, the observer acts as a complete participant. The most important feature of this approach is that it enables the study of *ad hoc* collaborative interactions, including their content. For example, the nature of the interactions among the software developers observed required that the observer be in a position to understand the jargon used by the observees in their everyday work, as well as the culture of the organization and the sub-culture of the team. Moreover, in order to observe situations that were as valid as possible, the observer had to adopt the standpoint of an insider, and thus, as discussed above, become an insider himself. However, to avoid the introduction of subjectivity, the data were also validated and analyzed by researchers who were not involved on-site.

2.2. Industrial environment observed

Our data were taken from audio-video recordings of the interactions of the members of a software development team in a large software development organization, which agreed to participate in our research. This team was composed of 12 individuals

(i.e. 1 project manager supported by 11 software developers) ranging widely in age, with different levels of schooling (from a Bachelor's degree to a Ph.D. in computer sciences and engineering), and individual experience ranging from 2 to 16 years in the field and from 9 months to 5 years of service in the company. The team is continually evolving in a large software organization with a highly standardized process comprising several thousand software developers in several countries around the globe. In spite of the size of the organization, the setting has the attributes of smaller organizations, as development is shared among several small teams of up to 15 members each, often located at a single site. Such an environment means that developers regularly need to cooperate with stakeholders in remote locations and in different time zones. Their need to cooperation stems from the fact that they are grouped by functional expertise, as well as by product area or software component, both of which are tightly integrated to form an enormous solution of millions lines of code from a number of different technologies. In this complex context, collaboration with a wide range of experts becomes essential.

2.3. Observation phases

The observations were gathered in two phases. The first, which we called the *ethnographic phase*, spanned several months, during which time the observer was integrated into the group as a regular employee, i.e. as a software developer in the team investigated. Not only did this period allow the observer to become familiar with the organization, its specific culture, and the jargon used by its members, but it also enabled him to plan the appropriate data collection method to be applied during the second period, which lasted two months.

The data collection method used during the observation phase was carefully designed. We selected audio-video recording for many reasons, among them the fact that we wanted to measure *ad hoc* behavior, which is unpredictable by nature. This means that our observations would have been difficult to record by simply writing them down, since notes are typically taken when the observer knows what to look for. There were other advantages to this method, one of which is that the recordings could be reviewed repeatedly in order to define and refine the specific attributes to look for in our raw data. This was helpful in our case, as several iterations were needed to define our data model. Another was that, unlike notes, recordings collect detailed information in ways that enable the analyst to move beyond real-time bias.

The issue of potential bias due to the presence of the camera was addressed by starting the recording sessions three weeks before our target date, in order to accustom our participants, who were not aware of this buffer period, to the presence of the camera. An indication that the behavior of the observees was not affected by the ever-present camera was that the subjects rarely looked in the direction of the camera once they had become used to it. During the two months of the observation and data collection phase, almost 200 hours of audio-video recordings of working sessions were collected.

2.4. Data model

Figure 1 presents the UML-based model used for the analysis of the raw data collected. This model is based on a combination of the linguistic and the object-oriented paradigms, as used by D'Astous and Robillard [4][16], and it defines observational units and the relevant attributes that should be observed in the raw data.

In Figure 1, the *working session* represents a particular timeframe during which observations on *ad hoc* collaborative activities are made. It can involve one *participant* or many, and their corresponding identifier and working start time and end time are recorded. As described by Kerbrat-Orecchioni [17], *Interaction* is the largest unit of observation being a communicative unit of one or more sequences which presents clear internal continuity, while being distinct from what precedes it and what follows it. There are no attributes associated with these interactions, since these can be derived from the underlying sequences. The *sequence* entity is the smallest unit analyzed in an interaction, in the form of a block of one or several exchanges between participants that show a high degree of semantic or pragmatic coherence, i.e. they are related to the same topic or oriented towards the same activity. Sequences represent a particular activity being performed on a particular topic.

Communication channels have different attributes. The Peer to Peer (face-to-face – F2F) sequence is characterized by an initiator, a start time, and an end time. It involves at least one other participant and may also involve a movement, so that the sequence could take place in a single cubicle. The phone sequence is characterized by a caller and a recipient of the call (a *callee*), and by a start and an end time. The e-mail sequence differs in that it comprises a sender as well as one or several receivers, a sent time, a message subject, a written size and a read size (in number of words),

and, finally, an index, if the message occurred within a thread of several messages.

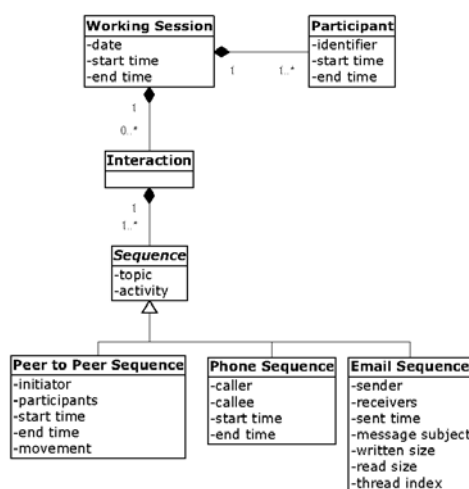


Figure 1. Data model designed for encoding the raw data

2.4. Ethical issues

Every means possible was employed to guarantee the ethical use of the recordings. Our research protocol was approved by an independent committee of the École Polytechnique de Montréal, which is mandated to supervise research undertaken with human subjects. They issued an Ethics Certificate for this research. All the subjects involved in our study were duly informed that their work sessions would be audio-visually recorded, as well as of the nature of the study. They all signed the letter of agreement required for certification. They were told that they were completely free to end their participation in the research at any time. Ethical issues were handled according to the established Canadian policies for research involving humans [15].

3. Data analysis

Detailed observations were made of the activities of four of the twelve development team members. These subjects were chosen through a judged sampling during our ethnographic phase, because they presented more team behaviors than the others within the context of the larger development team. They were closely co-located, their tasks were related to the same projects, and they were used to working together.

3.1. Recorded working sessions

From this first sample, a number of the working sessions analyzed were removed in order to discard bias due to non-representative sessions, where some of the subjects were absent or involved in formal meetings, preventing them from collaborating with their colleagues in the usual way. In accordance with the social tradition approach, we analyzed typical working sessions, which are defined as sessions where all four observed subjects are present and which last half a day. These sessions are not disrupted by occasional events, such as visitors, meetings, or special duties to be performed by one of the team members.

Finally, around 35 hours of audio-video recordings were analyzed, in which a total of 404 *ad hoc* F2F collaborative interactions were observed involving any of the four subjects interacting with another subject, as well as with other stakeholders.

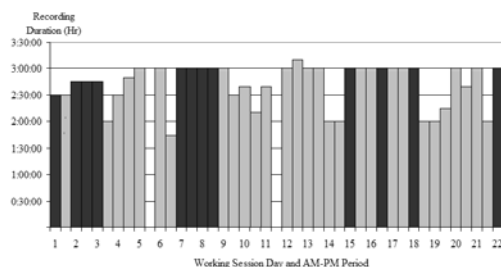


Figure 2. Recorded working sessions

Figure 2 shows the 42 sessions recorded over 22 days at a rate of 2 periods per day (AM: numbered column; and PM: unnumbered column). The recording was continuous during a session, and the recording sessions lasted 2 hours and 30 minutes on average. The 12 sessions used in this study are highlighted in a darker color. It is worth noting that the sessions analyzed spread over the 22 days of the recording periods. Two series of sessions are consecutive (2AM-2PM-3AM and 7AM-7PM-8AM-8PM). There were two half-days when no recording took place (5PM and 11PM).

3.2. Measurement of *ad hoc* collaborative activities

Figure 3 shows the percentages of *ad hoc* collaborative activities performed by the four observed participants. They took part in informal, spontaneous

and *ad hoc* F2F collaboration 28% percent of the time. Telephone *ad hoc* interactions occupied only 1% of the time, indicating that this channel was not significant in the setting studied. As a result, the following figures do not take into account telephone interactions.

This pie chart confirms the importance of the phenomenon revealed in previous studies [2,3]. In our case study, the observed participants spent an average of almost 2 hours and 20 minutes per 6 hour observed workday, which is considerably more than the 75 minutes observed by Perry *et al.*, but less than the average of 3 hours and 15 minutes observed by Robillard *et al.*

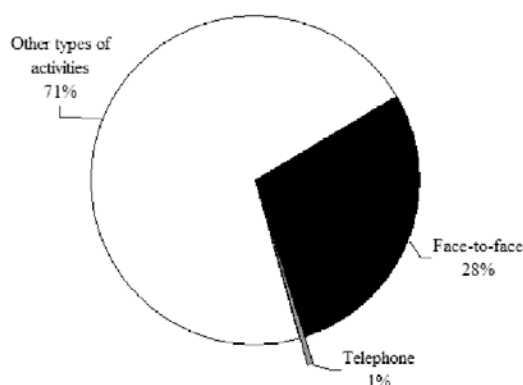


Figure 3. Percentage of time spent on *ad hoc* collaborative activities

As a side note, while Perry *et al.* collected their data from direct observation, Robillard *et al.* compiled their results from logbooks filled in by the participants themselves. However, even though the different measurement approaches could explain a portion of the observed variations between the three studies, they could certainly be attributable to the different settings in which the studies were conducted, in spite of the fact they were all conducted in the industrial world.

Figure 4 shows the average percentage of *ad hoc* interactions observed in terms of time spent during a session and for each of the 12 recorded sessions. It illustrates importance of *ad hoc* interactions and the large variations in *ad hoc* activities among sessions. We may recall that *ad hoc* interactions are spontaneous, unplanned, and of unknown duration. *Ad hoc* activities may be seen by participants who are not initiators of the interactions as an intrusion, leading to an interruption in what they are doing.

In Figure 4, some sessions, like 6 and 11, involve little activity, while others (1, 2, and 8) involve almost

40% of the total activity of the session. It has been observed that there is no specific pattern of activity. This distribution suggests the spontaneous nature of *ad hoc* activities.

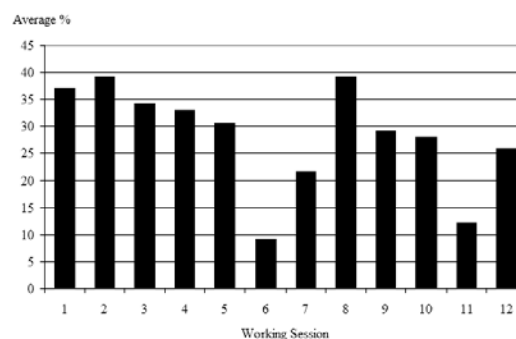


Figure 4. Average percentage of time spent on collaborative activities for each session per subject

Figure 5 illustrates the duration distribution in minutes of the 404 *ad hoc* collaborative interactions recorded during the 12 observed sessions. Almost 40% of the *ad hoc* interactions lasted less than a minute, and fewer than 4% of the *ad hoc* interactions lasted more than 15 minutes. This tends to confirm the accepted practice of allowing spontaneous interruptions for team members to answer their team mates' questions, knowing that they will not last very long and that the team members will shortly resume their own activities.

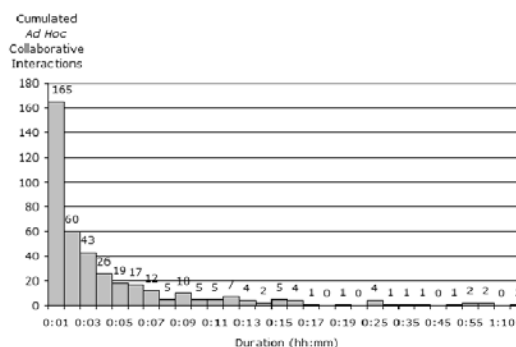


Figure 5. Cumulative numbers of *ad hoc* collaborative interactions as a function of their duration in minutes

3.3. Role definition

The observations of the team mates collected at the beginning of the study enabled us to identify the role of each subject within the team dynamics. These roles are often informal and implicit, and not formally acknowledged by the main players. Based on qualitative observations, the roles of the four subjects identified are the following:

- MS1: the new recruit on the team
- MS2: the project manager, who occupies the formal leadership position
- MS3: the individual responsible for configuration management of the software built by the team (informal leadership, rooted mostly in his knowledge and expertise)
- MS4: no specific role (can be seen as the embodiment of an average developer)

Figure 6 underlines the main characteristics of the four subjects and their respective roles. The size of the bubbles represents the relative proportion of all the interactions in which each subject was involved. For example, MS3 is the participant having the most *ad hoc* interactions, while MS1 is the one with the fewest *ad hoc* interactions.

The size of the arrow indicates the relative number of interactions initiated by the subject from whom the arrow originates, and the relative number of interactions directed to the subject towards whom the arrow points. The darker arrows indicate the leading direction for each communication channel. For example, MS1 initiated 10 *ad hoc* interactions with MS2, while MS2 initiated 5 *ad hoc* interactions with MS1.

Arrows also point from and towards people outside the group of four observees. We recall that the four observees are part of a team of twelve. So, eight more individuals directly interact with the observees (labelled L). There are also eight other individuals (coincidentally, the same number), who are from another group and who interact with the observees (labelled D). Each arrow pointing from or towards these two groups refers to the total number of *ad hoc* interactions with any one of these sixteen individuals. For example, MS1 initiated 30 *ad hoc* interactions with someone on the team other than the four observees, and 24 *ad hoc* interactions were directed to MS1 originating from someone other than the four observees.

Figure 6 presents in detail the *ad hoc* interaction patterns between the observees and the people outside the team. MS1 extracts information from the team. He mostly initiates *ad hoc* interactions to obtain

information, as is shown by dark arrows leaving MS1 and the fact that he is rarely involved in responding to requests from the team observees. This is a quantitative demonstration of the impact of a newcomer on the team dynamics.

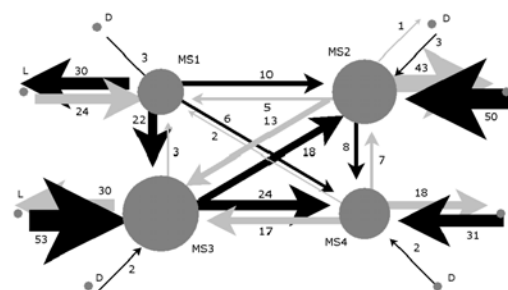


Figure 6. Network of *ad hoc* collaborations between the subjects and other stakeholders

MS4 experiences the opposite situation, as all the dark arrows are pointing towards him, which indicates that people are asking him questions. He is more often interrupted by *ad hoc* interactions than he is initiating them. This pattern quantitatively confirms a cooperator role on the part of MS4. He collaborates in peer-to-peer relationships with his colleagues, even though he is not involved in a significantly higher percentage of interactions than the new recruit, MS1, as shown by the relative size of the MS4 bubble.

MS3 displays an interesting pattern: although queries are usually referred to him by someone outside the organization, as represented by the large dark arrow indicating a total of 53 incoming *ad hoc* interactions, he himself often refers them to MS2 or MS4. We may recall that MS1 is a junior team member and, as such, he is not likely to be very helpful to MS3. These data quantify the importance of the reference role in the team dynamics, in this case, to the extended team.

MS2 is another main *ad hoc* communicator with people of the team. The number of MS2 *ad hoc* interactions incoming from rest of the team (L) is as large as the number outgoing from MS2. However, these queries are not passed along to the other observees, as was the case for MS3. We may recall the formal leadership role of MS2 in his position as project manager, and the informal leadership role of MS3 as a source of information and knowledge for his team mates. Accordingly, Figure 6 constitutes a quantitative demonstration of the central place of leaders on a team and the importance of the *ad hoc* interactions associated with this leadership.

We observed that face-to-face *ad hoc* interactions can occur in an active or in a verbal mode. The active mode involves movement, where a team member physically visits another team member's cubicle. Team members are in verbal mode when they are close enough to verbally interact without leaving their workstations. Verbal mode activity is dependent on the physical setup of the working environment. Figure 7 shows the interaction network pattern that corresponds to the verbal mode. This pattern is a subset of the pattern in Figure 6.

From Figure 7, it can be seen that MS4 is now the main actor in the verbal *ad hoc* interactions. MS4 occupied a central cubicle in the office layout. It can also be seen that there is some equilibrium between the initiations of verbal interactions. For example, MS4 initiated almost as many verbal interactions as he received from each of his neighboring participants. Figure 7 provides a quantitative demonstration of the impact of the physical position of a member in the team's *ad hoc* collaboration dynamics.

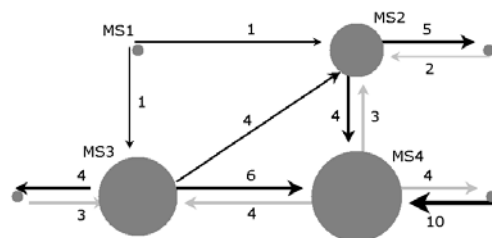


Figure 7. Network of *ad hoc* collaboration between the subjects in verbal mode

4. Study limitations

Most studies involving developers in real working environments are based on think-aloud protocol analysis or observer-recorded events, often based on a single individual. These studies are characterized by detailed observations of short event duration. Our study is characterized by long-term non-invasive recording of team behavior in a professional environment. The videos enable us to record all the movements and most of the discussions going on among the team members.

We have been able to derive an accurate pattern of team behavior related to face-to-face interactions. We are aware that the physical setup is of paramount importance in such behavior. The physical environment was designed to facilitate face-to-face interaction. The cubicle layout was attractive, with an open seating area

and short walls separating the cubicles. In this open and friendly environment, internal telephone calls were rarely used. One of the participants, now working in another firm where the environment is less open, is using the telephone much more.

Our study is based on the activities of a team involved in the maintenance of a large software product used worldwide. The managerial style may also have an effect on the behavior of the team members. All these factors, and many more, may have an impact on the observed results.

The purpose of this research is to understand the role of *ad hoc* interactions in a software development organization. We found that this aspect of software development is very important in terms of the frequency of occurrence and the duration of these interactions during a working session.

Studies are currently under way to analyze the content of these interactions. Face-to-face interaction is also being studied in connection with the use of e-mail as a means for communicating information.

We believe that this study, although conducted in a specific professional environment, can provide some interesting information to enable a better understanding of the needs of face-to-face communications in a software development team.

5. Concluding remarks

This paper presented an empirical study conducted in an industrial software engineering setting with the objective of measuring and analyzing *ad hoc* collaborative activities. Observational approaches borrowed from the social sciences have enabled us to quantitatively measure the activities sustaining *ad hoc* interactions.

This work is based on an exploratory case study, and attempts to provide a deeper understanding of the process underlying face-to-face exchanges. It focuses on finding and quantifying the parameters that characterize the exchange process.

Such studies are important for many reasons. First, they provide a deeper understanding of human activities within a software development team, enabling us to better understand and quantify the various roles of the participants.

Second, they provide some quantitative basis for evaluating the cost of *ad hoc* interactions with respect to the various roles. *Ad hoc* interaction is akin to "just-in-time" learning.

One of the major results of this study is the quantitative evaluation of *ad hoc* collaboration, which

requires a great deal of time on the part of the team members and is rarely taken into account in task assignment. Moreover, there is a relationship between the roles people play within a team and the time they spend in *ad hoc* collaboration activities.

There is also a relationship between the physical space that the team occupies and their verbal *ad hoc* activities. This study suggests that, in an open office, the guru should be centrally located.

Ad hoc collaboration enables people to share their knowledge or validate their understanding in a just-in-time paradigm. Face-to-face communications seem very important, as illustrated by how little e-mails and other electronic media available for communicating are used (this research is in progress and data are not presented in this paper). The social aspects of face-to-face *ad hoc* collaboration have not yet been explored, but we suspect that it is fairly important and that people need human contact and to use body language to express themselves.

Although this study was performed in a software development environment, we feel that face-to-face behavior may be of broader interest. We speculate, therefore, that any teamwork based on creativity and where the exchange of knowledge is important will be characterized by similar behavior. The major characteristic of face-to-face behavior revealed in this study is the opportunistic nature of the exchanges, which is associated with the creative aspect of the tasks to be performed. Another characteristic is the link between this behavior and the roles of the team mates, which are not unique to the software development environment. Newcomer, guru, and manager alike are likely to behave in the same way, regardless of the project domain. These findings are probably valuable in any environment where the tasks to be performed are intellectual and rely on the sharing of knowledge. However, the physical setup, which is an open-plan office here, may have a measurable impact on the exchange patterns.

The next step in this research project is to analyze the content of the *ad hoc* collaboration. The sharing of information is the basis for teamwork. However, preliminary studies indicate that it is only one of the objectives of *ad hoc* collaboration, others being cognitive synchronization, problem solving, and coordination, for example.

This study raises many questions. *Ad hoc* collaboration is very time-consuming, since a team member's work is interrupted to listen to a team mate's request. So, for example, would it be efficient to restrict *ad hoc* collaboration within the workplace?

Software development is becoming so complex that team activities are required on most projects. Team activities based on human interaction are a requirement for the emergence of new ideas and designs. The social sciences have proven sets of methods and research paradigms for studying human interactions.

This paper illustrates how these approaches could be used to derive the exchange patterns of *ad hoc* interactions in a team project. Patterns and theories emerging during the course of the research and reported here will now be validated by confirmatory research using these approaches.

References

- [1] Q.R. Jett, and J. M. George, "Work interrupted: A closer look at the role of interruptions in organizational life", *Academy of Management Review*, vol. 28, 2003pp. 494-507.
- [2] D.E. Perry, N.A. Staudenmayer, and L.G. Votta, "People, organizations, and process improvement", *IEEE Software*, vol. 11, no. 4, 1994, pp. 36-45.
- [3] P.N. Robillard, and M.P. Robillard, "Types of collaborative work in software engineering", *Journal of Systems and Software*, vol. 53, no. 3, 2000, pp. 219-224.
- [4] P. D'Astous, and P.N. Robillard, "Empirical study of exchange patterns during software peer review meetings", *Information and Software Technology*, vol. 44, no. 11, 2002, pp. 639-48.
- [5] J.D. Herbsleb, and R.E. Grinter, "Splitting the organization and integrating the code: Conway's law revisited", *Proceedings of the 21st International Conference on Software Engineering (ICSE '99)*, 1999, pp. 85-95.
- [6] S. Cherry, and P.N. Robillard, "The social side of software engineering: A real *ad hoc* collaboration network", *Int. J. Human-Computer Studies*, Vol. 66, 2008, pp. 495-505.
- [7] C.B. Seaman, and V.R. Basili, "Communication and Organization: An Empirical Study of Discussion in Inspection Meetings", *IEEE Transactions on Software Engineering*, vol. 7, 1998, pp. 559-572.
- [8] C. Gutwin, R. Penner, and K. Schneider, "Group Awareness in Distributed Software Development", *Proceedings of the 2004 ACM Conference on Computer Supported Cooperative Work*, 2004, pp. 72-81.
- [9] T.D. LaToza, G. Venolia, and R. DeLine, "Maintaining Mental Models: A Study of Developer Work Habits", *Proceedings of the 28th international conference on Software engineering*, 2006, pp. 492-501.
- [10] J. Chong, and R.Siino, "Interruptions on software teams: A comparison of paired and solo programmers", *Proceedings of the 20th Anniversary ACM Conference on Computer Supported*

- Cooperative Work*, 2006, pp. 29-38.
- [11] A.J. Ko, R. DeLine, and G. Venolia, "Information Needs in Collocated Software Development Teams", *Proceedings of the 29th international conference on Software Engineering*, 2007, pp. 344-353.
 - [12] P.M. Sanderson, and C. Fisher, "Exploratory sequential data analysis: foundations", *Human-Computer Interaction*, vol. 9, no. 3-4, 1994, pp. 251-317.
 - [13] E.R. Babbie, *The practice of social research*, Belmont, CA : Wadsworth Thomson Learning, 2001.
 - [14] D.L. Jorgensen, *Participant observation: a methodology for human studies*, Newbury Park, Calif.; London : Sage Publications, 1989.
 - [15] NSERC, Natural Sciences and Engineering Research Council of Canada. Tri-Council Policy Statement: Ethical Conduct for Research Involving Humans. In. *Interagency Advisory Panel on Research Ethics*. 2005.
 - [16] P. D'Astous, and P.N. Robillard, "Characterizing implicit information during peer review meetings", *Proceedings of International Conference on Software Engineering*, 2000, pp. 460-466.
 - [17] Kerbrat-Orecchioni, C., *Les interactions verbales*, Paris : Colin. 1998.

Annexe H

Importance of Peer-to-Peer *Ad hoc* Interaction in the Development of Large Software Systems

***Actes de l'Ergo'IA 2006 - L'humain comme facteur de
performance des systèmes complexes***

Bidart-Biarritz, France, 2006

Importance of peer-to-peer *ad hoc* interaction in the development of large software systems

Sébastien Cherry

Département de génie informatique
École Polytechnique de Montréal
C.P. 6079 succ Centre-ville
Montréal, Qc. Canada, H3C 3A7
Sebastien.cherry @polymtl.ca

Pierre-N. Robillard

Département de génie informatique
École Polytechnique de Montréal
C.P. 6079 succ Centre-ville
Montréal, Qc. Canada, H3C 3A7
Pierre-n.robillard@polymtl.ca

RESUME

Le développement de logiciel comporte une activité cognitive importante. Les membres des équipes de développement communiquent souvent entre eux de façon *ad hoc* par une interaction spontanée de personne à personne ou encore par courriel. On observe un intérêt grandissant pour la conception d'outils capable de faciliter le développement distribué de logiciels sur différents sites. On connaît, toutefois, relativement peu la dynamique et l'efficacité des communications entre des personnes qui travaillent ensemble sur un même projet. Cette étude empirique explore les aspects de la communication *ad hoc* entre intervenants dans le cadre d'un travail d'équipe de développement logiciel en milieu industriel. Les observations sont effectuées à partir d'enregistrements audio-vidéo des activités des membres de l'équipe. Les méthodes d'analyse des données sont empruntées aux domaines des sciences sociales. Les caractéristiques des communications de personne-à-personne sont présentées. On constate que ces communications ad-hoc occupent plus de 30% du temps des équipes et que le rôle des communications par courriels est secondaire à l'intérieur de l'équipe. Différents profils de communications sont présentés et comparés. Les patrons de communications ad-hoc globaux interne est externe à l'équipe observée sont également présentés.

KEY WORDS: *Ad hoc* interactions, peer-to-peer communications, e-mail messaging, empirical studies, team interactions, communications profiles.

ABSTRACT

Software development is an intensive cognitive task where co-located team-mates experience peer-to-peer *ad hoc* interactions and e-mail messaging. Global software developments are oriented toward the use of interactive software tools to support and increase the efficiency of globally distributed software developments. This empirical study explores the mechanisms of *ad hoc* human in-

teractions in a complex industrial software development environment. Observations based on audio-video recording are analyzed with methodology from the human sciences research domains. The observed characteristics of peer-to-peer *ad hoc* activities are presented. It shows the impact of the participant roles on the *ad hoc* communications profile. The needs for *ad hoc* peer-to-peer interactions are very important in co-located team and could represent up to 30% of the total team time resources. E-mail interactions were found relatively less important. The various profiles of *ad hoc* communications are compared. Global patterns of interactions are presented.

INTRODUCTION

Software engineering is nowadays more and more recognized as a human experience, even though human aspects involved in software development have been little studied in the past. The whole system is quite complex since the human are interacting with computer through various software tools and the human are interacting between them to share information and to synchronize their mental model of the system to be designed. With the raise of global software development many software tools are being built to facilitate distributed development and team interaction. We realized that we know little about human interactions in co-located teams. This interaction is essentially *ad hoc* and based on peer-to-peer meetings or e-mail exchanges.

This neglect, as stated by Perry Staudenmayer and Votta [8], owes primarily to the difficulty of measuring quantitatively the human facets of engineering. However, as observed by Seaman [11], more and more studies have aimed to measure "people factors". Among these factors, communication [12], coordination [4,5] as well as collaboration [10] between software engineers are relevant aspects on which several researchers have focused in the past few years.

Following in the stride of these studies, Perry, Staudenmayer and Votta, one of the pioneers in the field, tried to describe how developers spend their time as a software development project unfolds [8]. Their studies showed, among other things, that developers spent 75 minutes per day on unplanned informal communications. This observation was later supported by Robillard and Robillard in their study on the different types of collaborative work in software engineering [10]. The types of works are mandatory meetings, called or scheduled meeting, *ad hoc* meetings and individual works. A later study by D'Astous and Robillard [3] analysed the exchange patterns between team-mates during called meeting. These meetings are scheduled in advanced and are mostly peer review meetings. They built a model to represent the qualitative and quantitative importance of various exchanges occurring during the peer review meetings.

Ad hoc meetings are more difficult to study given their spontaneous nature. They are nevertheless important since they account for a significant amount of time spent on a software project as observed by Perry and his colleagues. These observations convinced them of the importance of these activities for team dynamics, a conviction shared by many more authors in subsequent studies. Herbsleb and Grinter [5] and Seaman and Basili [12] come to mind. Despite a fairly large consensus, no known study yet describes and characterizes *ad hoc* collaborative activities and their corresponding communications patterns.

These studies are important to measure the impact of new technologies such as e-mails and chats on the efficiency of team interaction in building large and complex software projects.

RESEARCH METHODOLOGIES

During the course of this research the observer act as a complete participant for several reasons, the most important being that the topic of this research, was to study the efficiency of *ad hoc* collaborative interactions, including their content [1]. The nature of the interactions between the observed software developers required that the observer be in a position to understand the jargon used by observes in their everyday work as well as the culture of the organisation and the sub-culture of the team [6]. Thus, in order to reach this level of understanding, the observer had to get involved fairly actively in the activities of the group. Additionally, in order to observe situations that were as valid as possible, the observer had to adopt the standpoint of an insider and thus, as discussed above, became himself an insider. However, to avoid subjectivity sneaking in, the data were validated and analyzed by researchers that were not involved with the team studied.

Our observations, based on audio-video recording, were made on a software development team who performed in a large software development organization. This team was composed of 12 individuals of a wide range of ages, of different degrees of schooling going from bachelor to doctoral grades in computer sciences and engineering, with individual experience ranging from 2 to 16 years of experience in the field and ranging between 9 months to 5 years of service in the company. In addition, even if this team evolves within a large software organisation, with a much standardised process, comprising several thousands of software developers spread over several countries around the globe, the setting comprises attributes of smaller organisations as the development is divided in several small teams of up to 15 members, often grouped in one single location. However, developers regularly need to cooperate with stakeholders from remote locations and different time zones because they are divided by functional expertise and are also segregated by product areas or software components that are tightly integrated with each other to form a enormous solution of millions of lines of code. In this complex context, collaboration with a wide range of experts becomes necessary.

Furthermore, the observation phase was conducted in two successive periods. The first one, which we called the *ethnographic period*, was performed over a time interval of several months during which the observer integrated the group as a regular employee.

Everything was put into place to guarantee the ethical use of recordings and our research protocol was approved by an independent committee at École Polytechnique de Montréal, which is mandated for supervising research with human subjects, and they delivered an Ethic Certification for this research.

Furthermore, the issue of potential bias due to the presence of the camera was addressed by starting recording three weeks before our desired date in order to make our subjects, who weren't aware of this buffer period, accustomed to the presence of the camera, thus reducing bias to a minimum.

The audio-video recording was combined with other methods, including the automatic copy of all electronic e-mail messages exchanged between the members of the group by means of the messaging software used in the company, a daily back-up of the source code of the software developed from the versioning system as well all the files found on the file server used by the team where working documents are stored. With the audio-video recordings and e-mail messages exchanged, we were confident we could capture some of the complexity of the cognitive behaviours of software developer in a complex environment. Then, during the two months of observa-

tion and data collection phase, we collected the following data:

- Up to almost 200 hours of audio-video recordings of working sessions;
- Around 2500 e-mail messages exchanged between team mates;
- daily back-up of the source code from the versioning system
- all documents found on the file server shared by the team.

This paper reports on the analysis of the peer-to-peer *ad hoc* interactions and the e-mail exchanges.

DATA VALIDATION

The reliability of such studies relies on the quality of the qualitative data recorded. Two agreement validation phases were held. The first was an intra-coder agreement, where a number of encoded data sequences were re-encoded a month later by the same coder, and the second was an inter-coder agreement, where the same operation was accomplished, but this time by another coder who was able to understand the context and the jargon as well as the primary coder. Then, to measure the degree of agreement between encoded and re-encoded data sequence, and to eliminate from the estimation the part of probability that the same code had been chosen by the two coders by chance, the use of intra- and inter-coder agreement coefficients and indices was made [2,9].

A detailed analysis was performed on data obtained from four of the 12 members of the observed team. These subjects were chosen because, through a judged sampling, they were identified during our ethnographic phase as presenting a much more representative behaviour than the others.

A stratified sampling of working sessions has been realized and from this first sample, a number of analyzed working sessions were removed in order to discard bias due to non-representative sessions where some of the subjects were absent or monopolized in formal meeting preventing them to collaborate with their colleagues as usual. Finally, up to around 35 hours of audio-video recordings were analyzed during which a total of 431 *ad hoc* collaborative interactions were observed involving our four subjects interacting with each other as well as with other stakeholders.

SUBJECT ROLES

The observations of the team-mates in their natural day to day activities prior to the beginning of the recording

enable us to identify the role of each of the four subjects within the team dynamics.

- The rookie: MS1; is the new recruit in the team. He tries to gather, consolidate and crystallize all pieces of information to understand the environment as well as the technical content involved in developing and maintaining software.
- The coach: MS2; is the project manager who occupies the formal leadership position. His interests are centered as much on the task as on the person, hoping for little more than adequate output while he tries to convince and motivate his subordinates by making occasional compromises.
- The reference: MS3; is responsible for configuration management. He is not to be considered as a guru, but he is a fount of knowledge for his colleagues who often ask him for information or help in order to solve problems.
- The cooperator: MS4; does not occupy any specific, formal or informal, role in the team. He is the embodiment of your average developer, performing his tasks well and cooperating fruitfully with his colleagues.

MEASUREMENTS OF AD HOC ACTIVITIES

Ad hoc interactions were identified as activity forming a logical communicative unit of one or more sequences that presents an evident internal continuity while being distinct from what precedes or follows as defined by [7]. We recall the very specific nature of these interactions, which are spontaneous, unplanned, and of unknown duration. *Ad hoc* activities can be seen by most of the subjects as an interruption of what they are doing.

Figure 1 illustrates the importance of *ad hoc* interactions. It shows the average percentages of *ad hoc* collaborative work observed for each subject during the total length of analysed recording.

For example, MS1 spend 21% of his total working time in *ad hoc* activities. The two subjects who exhibit one form or another of leadership (MS2 and MS3) spend more time on *ad hoc* collaboration than the other two subjects. The coach and the reference roles spent 29% and 41% respectively of their total time on *ad hoc* collaborative activities. As for the rookie MS1 and the cooperator MS4, they spend an average of 21% and 25% percent, respectively.

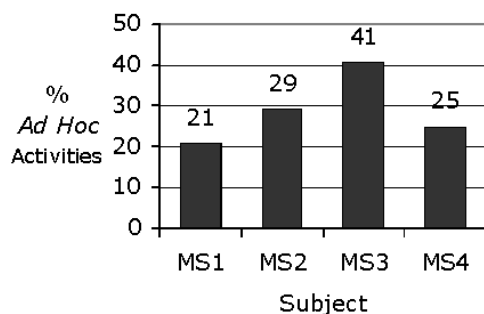


Figure 1. Average percentages of *ad hoc* collaborative work observed per subject

On a team basis, the average *ad hoc* activities accounts for almost 30% of all of their activities. *Ad hoc* interactions are only one component of the collaborative activities. The other two main collaborative activities are scheduled meetings and mandatory meetings. Teammates are also doing individual work.

Ad hoc activities must be initiated by someone. Figure 2 shows percentage of interactions initiated by each subject. For example MS1 has been involved in 108 *ad hoc* interactions from which he initiated 69, which sums up to 64%. The rookie, MS1 initiates almost twice as much *ad hoc* interaction as the other team members. This is coherent since the rookie is new in the team and on his way up the learning curve. His efficiency depends on gathering more information than the other players. This data may quantify the saying that adding someone new to a project that is late will just make the project later since many *ad hoc* interactions are initiated by the newcomer to catch up with the team expertise.

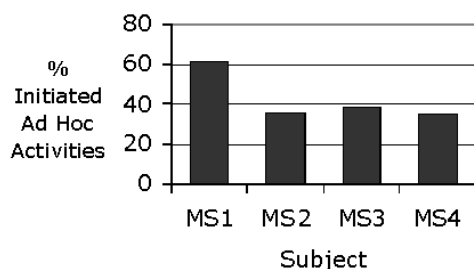


Figure 2 Percentage of *ad hoc* interactions initiated by the subject.

The four observed subjects are working as a team on few components of a large project. In the course of their tasks they need to interact with the 12 other participants of the

larger group and also with three people from outside this larger project group. Reciprocally, people from outside the observed team need to interact with them.

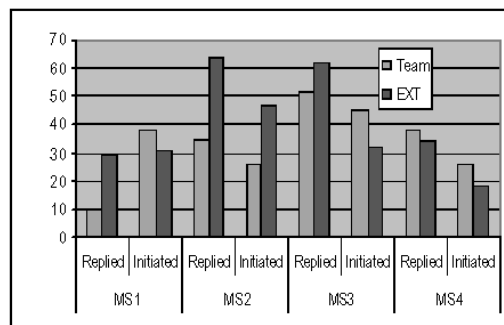


Figure 3. Number of replied and initiated *ad hoc* interactions internal and external to the team.

Figure 3 presents the number of *ad hoc* interactions internal and external to the team for each observed subject. For example, looking at the second pair of columns starting from the left, we see that MS1 initiated almost 40 *ad hoc* interactions with his team-mates, while he replied to almost 30 *ad hoc* interactions from someone outside the team. We observed that there is almost as much interaction between the team-mates than with people external to the team. MS2 which plays the role of the manager, is the main link for *ad hoc* communications coming from the outside, as expected.

The internet and specifically the e-mail is an important tool for facilitating written *ad hoc* interactions particularly in software development environment. However, there is an important difference between peer-to-peer *ad hoc* interaction and e-mail messaging. Peer-to-peer *ad hoc* interaction is always synchronous while e-mail communications could be asynchronous.

We studied the exchange of e-mail messages to see how this communication mechanism compares with peer-to-peer *ad hoc* interactions within this project environment.

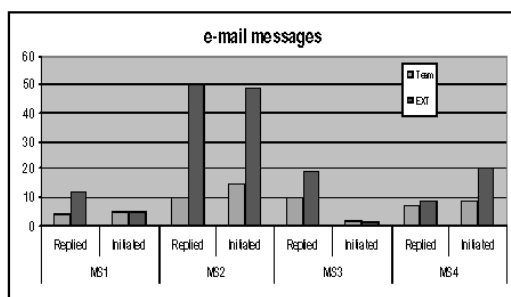


Figure 4. Number of replied and initiated e-mail messages internal and external to the team.

Figure 4 shows that e-mails were not intensively used within the team members. They seem to prefer peer-to-peer communications. However e-mail is extensively used by MS2, the manager, to communicate with the external team members and reciprocally they used the e-mail to communicate with him. Comparing Figure 3 and 4, we see that MS2 is the only team member with as much e-mail as peer-to-peer interactions.

PATTERNS OF AD HOC EXCHANGES

Ad hoc interactions, peer-to-peer or e-mails, are an important component of co-located team dynamics. In the following we integrate the data from each participant to derive general patterns of *ad hoc* activities.

Ad hoc interaction profile peer-to-peer and e-mail

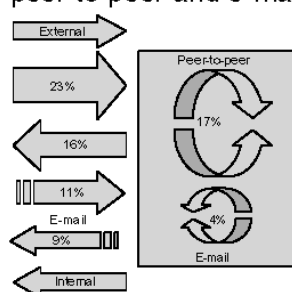


Figure 5. Overview of peer-to-peer and e-mail ad hoc interactions.

The box in the right hand side of Figure 5 represents the team activities composed of the four observed subjects. Inside the box we represent the relative *ad hoc* interactions internal to the team. The upper round arrows indicate that 17% of the total ad hoc interactions are peer-to-peer among the team members, while the lower round arrows indicate that 4% of the total team interactions are via e-mails.

The *ad hoc* activities involving people outside the observed team are represented by the straight and broken arrows on the left of the team box. The tail of the arrow indicates the origin of the *ad hoc* activities. An arrow originating from the left and pointing to the box represents *ad hoc* activities that are initiated from people external to the team. An arrow originating from the team box and pointing toward the left to the outside represents *ad hoc* activities that are initiated by one of the four team members. Straight arrows are for peer-to-peer interactions while broken arrows are for e-mail interactions.

The upper arrow pointing right toward the team box indicates that 23% of the *ad hoc* peer-to-peer interactions with one of the team-mates were initiated from an external team member. The second upper arrow pointing left away from the team box indicates that 16% of the *ad hoc* peer-to-peer interactions with external member were initiated by one of the team-mates. The second from the bottom broken arrow pointing toward the team box indicates that 11% of the *ad hoc* e-mail interactions were initiated from an external team member. The lower broken arrow pointing left away from the team box indicates that 9% of the *ad hoc* e-mail interactions were initiated from one of the team-mates to the people external to the team.

We see that peer-to-peer interaction is still the dominant means for people to interact within the team and with people outside the team. We recall that this study is done on co-located team and that the dynamic of *ad hoc* interactions could be quite different with distributed teams.

Fig 6 combines several of the results that were presented in the form of a path analysis of the *ad hoc* collaboration network between the subjects and other stakeholders. This graphic underlines the main characteristics of the four subjects and their respective roles. The size of the bubbles represents the relative proportion of all interactions in which each subject was involved, as shown before. The size of the arrows depicts the relative number of interactions initiated by the subject from whom the arrow is originating and directed to the subject towards whom the arrow is directed. For example, the communication channel MS1-MS2 has 13 communications through MS2 and 8 through MS1. Arrows are also pointing from and toward the outside team. Outside team is composed of 13 other individuals in the organization that interact with the observed subjects. Each arrows pointing from or toward the outside team is referring to the total *ad hoc* interactions with any one of the 18 persons. For example, MS1 initiated 36 *ad hoc* interactions with someone outside the team of the four observed subjects and 41 *ad hoc* interactions were coming from someone outside the team.

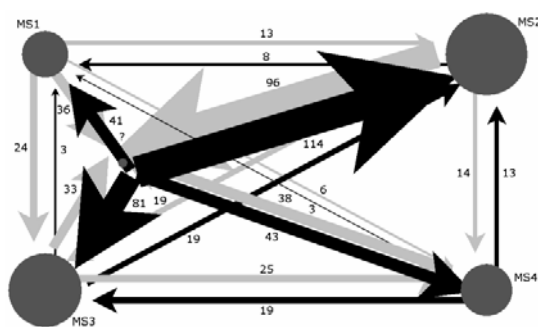


Figure 6. Path analysis of the *ad hoc* collaboration network between the subjects and other stakeholders

Figure 6 presents, in detail, the *ad hoc* interaction patterns from within the observed subjects and with people outside the team. The rookie (MS1) is absorbing information from the team. He is mostly initiating *ad hoc* interaction to get information as it is shown by the fact that large arrows are leaving MS1 and he is rarely involved in answering requests from inside the observed team. This is a quantitative demonstration of the impact of a newcomer on the team dynamic.

The cooperator (MS4) is experimenting the inverse situation where many large arrows are pointing toward him, which indicates that people are asking him questions. He is more often interrupted by *ad hoc* interactions than initiating *ad hoc* interactions himself. This pattern confirms quantitatively his cooperator status. He is exchanging information in peer-to-peer relationships with his colleagues even though he is not involved in a significantly higher percentage of interactions than the rookie.

The reference (MS3), is showing an interesting pattern. He seems to transfer the information from the outside world into the team. There is a large flow of information coming in from outside the team and this flow is propagated, through him, into the team toward MS2 and MS4. We recall that MS1 is a junior team member so he is not likely to have answers to the questions emerging from MS3. This data quantify the importance of the reference role within the team dynamic.

The coach (MS2) is the main *ad hoc* communicator with people outside the team. It is interesting to note that the input from outside the team is almost as important as the one for MS4 but in this case there is no pattern of internal propagation of the information. MS4 probably processes the information himself. His *ad hoc* interactions with the other team members are more balanced.

CONCLUSION

This paper presented an empirical study conducted in an industrial software engineering setting with the objective of measuring and analysing the complexity of the *ad hoc* collaborative activities observed in a professional software development team. *Ad hoc* interactions have been observed by previous studies but up until now, little published research has characterized this type of activities even though they can account for a fairly large proportion of the time spent on a software project. Observational approaches borrow from the social science enable us to measure quantitatively the activities sustaining *ad hoc* interactions.

Such studies enable us to uncover the importance of peer-to-peer interactions in team activities. This study was done in a high technological environment with expert in software engineering. Even so, it is interesting to see that e-mail is not the main channel for communications. Personal contact is an important parameter in team activities. Seen from the outside the team seem to be having a good cohesion and can work on this project by itself. This analysis shows that although the team has good internal cohesion it is interacting as much with the outside world as internally.

The needs for interactions in a complex environment are shown to be very important. The ergonomic of the working space for software development project should take this into account.

This study provides some quantitative basis for evaluating the cost of *ad hoc* interactions with respect to the various roles. The reference subject plays an important role in being a channel of communication between the team and the other people. An important part of its time is devoted to this activity. The manager should take this into account in planning the reference subject tasks.

This analysis stresses the importance of participant roles in *ad hoc* interactions. New team members need to reach team level expertise and to do so they need to communicate on an *ad hoc* basis even if formal training is available. *Ad hoc* interaction is kind of 'just in time learning'. Some people are friendly and collaborate with everyone or are kind of a guru and are resources for all. The cooperator subject of this study had somehow this pattern. The cooperator is important in team work because of his knowledge or his ability to help others. Undue pressure on him for task completion is likely to affect some of the team members since a number of subjects rely on him. It is acknowledged that managers are fairly busy. In this study we can see that his *ad hoc* interactions with outside the team are more important than with observed team-mates.

Software development is becoming so complex that team activities are required in most projects. Team activities based on human interaction are the requirements for the emergence of design and new ideas. Social sciences have proven sets of methods and research paradigms to study the human interactions. This paper illustrates how these approaches could be used to derive the exchange pattern of ad hoc interaction in a team project.

It provides a deep understanding of the human activities within a software development team. We can better understand and quantify the various roles of participants. Although this case study is based on the observation of a specific organization that constitute the primary limitation of this type of research, the results can be extrapolate at some extent to a broader set of context such as any software development organizations or eventually to any team setting with similar dynamics.

ACKNOWLEDGEMENT

This research would not have been possible without the agreement of the company in which it was conducted, and without the generous participation and patience of the software development team members from which the data was collected. To all these people, we extend our grateful thanks. This research was supported in part by NSERC grant A-0141.

BIBLIOGRAPHIE

1. Babbie, E. R. The practice of social research, Belmont, CA : Wadsworth Thomson Learning, 2001.
2. Brennan, R. L. and Prediger, D. J., Coefficient Kappa: Some Uses, Misuses, and Alternatives Educational and Psychological Measurement, vol. 6, pp. 687-699, 1981.
3. d'Astous, P. and Robillard, P. N., Empirical study of exchange patterns during software peer review meetings Information and Software Technology, vol. 44, no. 11, pp. 639-48, Aug 15, 2002.
4. Grinter, R. E., Herbsleb, J. D., and Perry, D. E., "The geography of coordination: dealing with distance in R&D work," Proceedings of GROUP 99: Conference on Supporting Group Work, 14-17 Nov. 1999, pp. 306-315, 1999.
5. Herbsleb, J. D. and Grinter, R. E., "Splitting the organization and integrating the code: Conway's law revisited," Proceedings of the 21st International Conference on Software Engineering (ICSE '99), 16-22 May 1999, pp. 85-95, 1999.
6. Jorgensen, D. L. Participant observation : a methodology for human studies, Newbury Park, Ca. ; London : Sage Publications, 1989.
7. Kerbrat-Orecchioni, C., Les interactions verbales Anonymous 1998. Linguistique. Paris
8. Perry, D. E., Staudenmayer, N. A., and Votta, L. G., People, organizations, and process improvement IEEE Software, vol. 11, no. 4, pp. 36-45, 1994.
9. Popping, R. On agreement indices for nominal data. In: Sociometric research, eds. Saris, W. E., Gallhofer, I. N., and International Sociological Association. Basingstoke : Macmillan, 1988. pp. 90-105.
10. Robillard, P. N. and Robillard, M. P., Types of collaborative work in software engineering Journal of Systems and Software, vol. 53, no. 3, pp. 219-224, Sep 15, 2000.
11. Seaman, C. B., Qualitative methods in empirical studies of software engineering IEEE Transactions on Software Engineering, vol. 25, no. 4, pp. 557-572, Jul, 1999.
12. Seaman, C. B. and Basili, V. R., Communication and organization in software development: an empirical study IBM Systems Journal, vol. 36, no. 4, pp. 550-563, 1997.

Annexe I

Communication Problems in Global Software Development:

Spotlight on a New Field of Investigation

Proceedings of The Third International Workshop on Global

Software Development (GSD 2004)

Édimbourg, Écosse, 2004

Communication Problems in Global Software Development: Spotlight on a New Field of Investigation

Sébastien Cherry, Pierre N. Robillard

Software Engineering Research Laboratory, École Polytechnique de Montréal
{sebastien.cherry, pierre-n.robillard} @polymtl.ca

Abstract

While it is widely recognized that communication plays a critical role in software development, it has been observed that problems of coordination may be generated when teammates in the field are working at a distance from one another.

This paper presents an ongoing empirical study on ad hoc collaborative activities which occur in an industrial software engineering environment. We believe that a better understanding of these types of activities and their content will pave the way to further solutions designed to enhance communication, and thus improve both collaboration and coordination in virtual software development settings.

We include details of our motivations for the study, followed by some methodological considerations, and, finally, some preliminary results which demonstrate not only the significance of our data, but also the relevance of our approach.

1. Introduction

Communication is undoubtedly an essential element which plays a critical role in a software engineering process in the gathering and crystallization [15] of all relevant information in quality software which fulfills the user needs on time and within budget [2], [3], [13], [14], [16]. Moreover, several studies have stressed the fact that informal communications seem to be fairly important in terms of the time spent on a software project. Perry, Staudenmayer and Votta (1994) found during a case study that informal communications take up an average of 75 minutes per day per software developer [12], and Robillard and Robillard established in another case study that ad hoc collaborative activities can occupy up to 41% of the developer's time [14]. Seaman (1996) [16] also supports the need for this type of communication if developers are to carry out their tasks adequately.

However, in the global software development setting, which has become a more common practice for many business reasons, some researchers have observed that communications, specifically informal ones, face

significant challenges by virtue of distance, both geographical and temporal [6], [7], [8], [9]. They note that the consequence of this is a potential for problems of coordination to occur.

This short paper presents an ongoing empirical study being carried out within the framework of a case study in the industry which explores the ad hoc collaborative activities that take place in a software engineering setting. By ad hoc collaboration, we mean all informal and spontaneous activities performed by two or more developers who are working on a particular task of the project. These activities can take many forms, such as in informal peer-to-peer conversations, also referred to in the literature as “water-cooler-talk” [6], as well as electronic mail exchanges and phone calls.

We believe that a better understanding of such activities and their content will pave the way for further solutions with the aim of enhancing communication and thus improving both collaboration and coordination in virtual software development settings. Our general approach is to observe and understand the informal and spontaneous collaboration activities that take place in a classical single-site software development environment where developers have as much freedom and opportunity to communicate as they wish, and to measure their positive and negative impacts on the rest of the software project. Depending on the results of our investigation, two avenues of action can be envisaged. The first might be to infer and formalize from our observations some state-of-the-art rules or practices applicable in global software development contexts which will be better adapted to the empirical reality and make collaboration between teammates working apart more efficient. The second might be to use this comprehension to give us some insight into the tools needed to support communications in a distributed software development environment with the intention of creating what some call the “virtual 30 meters” [9].

In this paper, we explain our research methodology in broad strokes and present some preliminary results which demonstrate not only the significance of our data, but also the relevance of our approach.

2. Research Methodology

Empirical studies have been becoming more and more popular in the past few years in software engineering, in parallel with the growing popularity of people-centered researches. Indeed, researchers must innovate in order to study this new topic of interest, namely people, by borrowing certain techniques used in the social sciences, such as psychology and sociology. Our research methodology has been inspired by several papers [4], [11], [18], [19] as well as aforementioned studies which have examined the human aspects of software engineering, but has been to some extent adapted to the field investigated in this study. Below, we give a general description of our research methodology.

2.1. Research Objectives

As stated above, good communication is the *sine qua non* of software development processes in obtaining quality products which meet user needs on time and within budget [2], [3], [13], [14], [16]. Research has shown the non-negligible importance of informal communications [12], [14], [16], while some has specifically highlighted the fact that distance in global software development is a challenge to informal communications which can generate problems of coordination [6], [7], [8], [9]. However, no research has described the content of this type of communication. These elements led us to define the following research objectives:

- To design a model of the ad hoc collaborative activities found in an industrial software engineering setting and characterize them, as well as to identify and describe the content of the communications that ensue.
- To generate a series of hypotheses emerging from the results of this research which could later be validated by confirmatory research.

2.2. General Approach

These objectives will be achieved by means of participant observation. This technique is suitable in exploratory research like ours where the goal is to inductively generate theories from direct observations, also called grounded theory, rather than to empirically verify a hypothesis formulated in advance [1], [10], [17].

2.3. Target-setting

The target is a team of eight individuals which develops software for commercial purposes in a large international company which has been in operation for

several years and which has a well-established software development process. It should be noted that, even though the observations are made in a large company, this last contains attributes of small or medium-sized organizations since the work is divided among small teams like the one that is participating in our research. Also, the members of this team are highly representative of developers in the industry, in terms of the wide variety of their ages, software development experience, schooling and length of service in the company.

2.4. Data Collection

The data collection phase, which lasted 8 weeks, is done. The methods used during this phase were selected following an earlier ethnography period of several months. The data collected during this period includes:

- 185 hours of audio-video recordings of working sessions
- 2496 electronic mails exchanged by the 8 teammates
- A daily backup of the source code and other artifacts found in the field

Audio-video recordings were preferred over field notes because they offer the enormous advantage that they can be consulted many times over. This is very important in exploratory research like ours where we do not necessarily know in advance what to observe. E-mails were automatically captured by triggers defined in the messaging software used in the company, and include both those received and those sent to allow cross-validation. Finally, a backup was made of the source code and artifacts found to allow further content analysis.

2.5. Data Analysis

The principal method used to analyze the large amount of data, mainly in the form of the audio-video recordings, was the Exploratory Sequential Data Analysis (ESDA) [5]. This method was chosen because it is particularly well-suited to exploratory research like ours, where theories have to be induced from empirical data, and, more importantly, where the sequential information of the data must be preserved.

Briefly, the ESDA process is iterative. It involves the definition, sometimes intuitive, of concepts arising from the ESDA tradition of taking the point of view of the researcher, subsequently providing a guide as to what to observe in the raw data and how to manipulate it to derive data on which theories will be founded. This process is done iteratively because it is often necessary to step back in order to add, remove or revise some concept definitions [5].

Of the eight different ways to manipulate data proposed by ESDA, encoding is certainly the most

important. This consists of labeling each data sequence with a code formed from an exhaustive, exclusive and limited list of categories. This allows qualitative data to be transformed into quantitative data, which in turn makes it possible to further manipulate the data, by performing statistical calculations, for example [5], [17].

3. Preliminary Results

The following results are based on observations made on the activities of four of the eight developers on the team over a period of 8 hours each. It does not take into account e-mail exchanges. Also, the four individuals observed were chosen because they have been seen to collaborate more with their teammates than the others. This choice is justified because our purpose is to study the content of the ad hoc collaborative activities that occurred, so that these particular individuals were simply more likely to give us more data to analyze.

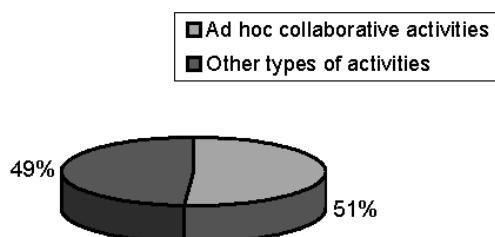


Figure 1. Distribution of time spent in ad hoc collaborative activities in comparison with other types of activities

As Figure 1 illustrates, 51% of the time spent on the project by the observed developers is occupied on ad hoc collaborative activities, in contrast with other types of activities. This is a surprising result which seems to strongly confirm the importance of the phenomenon observed in the target setting, but which needs to be validated by further analysis on a much larger scale.

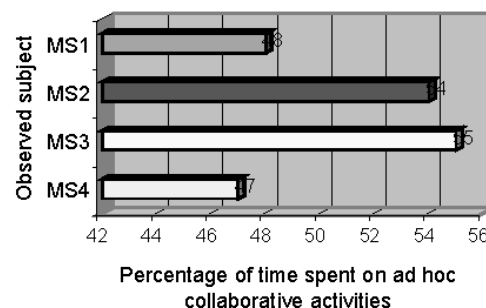


Figure 2. Percentage of time spent on ad hoc collaborative activities by observed subject

Figure 2 shows the time spent on ad hoc collaborative activities per observed subject. It can be noted that the percentages associated with subjects MS2 and MS3 are slightly higher than the others. This difference can possibly be explained by the nature of the work performed by these subjects since MS2 is the team's project manager and MS3 is in charge of the infrastructure for the software built and often the problem-solver on the team. An interesting thing to note is that, in 78% of the interactions in which MS2 is involved, his interlocutors initiated the interactions. However, it has been noted in the field that, most of the time, MS2 shares information by broadcasting a message to his team via e-mail instead of in peer-to-peer conversations. Thus, it will be interesting to analyze these e-mail exchanges. By the way, in each of the 82 interactions observed, an average of 2.3 stakeholders took part, and their average duration was 6:31 minutes.

As for the ad hoc collaboration activities observed, a preliminary outline has emerged from the raw data containing six categories, as follows: "cognitive synchronization" exists when two or more developers exchange information to ensure that they share the same knowledge or the same representation of the object in question; "problem resolution" occurs when two or more developers are aware of the existence of a problem and attempt by various means to solve the problem or to mitigate it; "development" occurs when two or more developers contribute to the development of a new feature or component of the software; "management" is the result of two or more developers coordinating and planning activities such as meetings, common working sessions or scheduling; and "conflict resolution" is the process of two or more developers taking part in discussions to resolve a difference of opinion. Ad hoc collaborative activities in the "not relevant" category, group together all the interactions that do not concern the project or the software built.

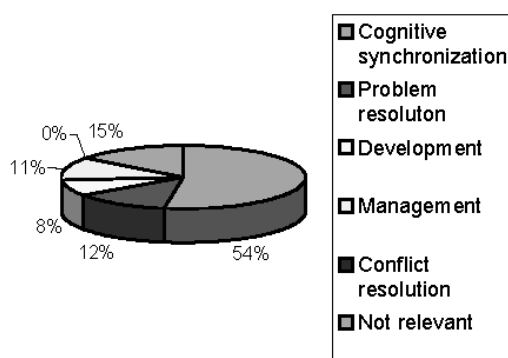


Figure 3. Distribution in number of occurrences of ad hoc collaborative activities identified

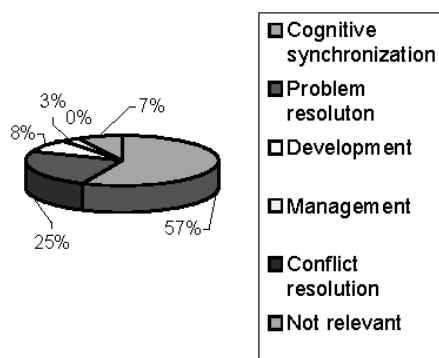


Figure 4. Distribution in terms of time spent on ad hoc collaborative activities identified

Figure 3 shows that, in a little over half the times when these interactions occur, they do so in the form of cognitive synchronization, and this tendency is supported by the data in Figure 4 which show the distribution in time spent. This is not a surprising finding, since it is well established that the sharing of information and knowledge is a crucial element in software development.

Moreover, it can be noted that problem resolution activities represent only 13% of the occurrences, but that, in terms of time spent, the percentage rises to 25%. This suggests that, when they occur, problem resolution activities take much longer than the others. This is supported by the statistics of time spent by sequence as function of ad hoc collaborative activities, which shows that a mean of 9:48 minutes is spent on problem resolution.

Finally, another interesting finding is that management activities, unlike problem resolution, represent 13% of the occurrences, but only 3% in terms of time spent. In other words, they are quite numerous relative to the small percentage of time spent on them. This result probably tends to support the theory of some researchers to the effect that informal communications are important in order that teammates can coordinate their activities efficiently [6], [7], [8], [9].

4. Conclusion

It is widely held that communication is a crucial element in software engineering, but, unfortunately, it is an aspect which seems to be lacking in global software development and one which must be addressed.

This paper presents an ongoing empirical study on ad hoc collaborative activities in an industrial software engineering setting. The general objective of this research is to gain a better understanding of these kinds of activities and their content in order to be able, subsequently, to propose software process enhancements with the aim of rendering collaboration between teammates more effective on the one hand, and, on the other, to obtain some insight into the tools needed to support communications in a distributed software development context.

We think that this kind of research is needed, first of all, because the importance of communication in distributed environments is poorly understood, but also to expose any wrong assumptions there may be that are often mistaken for the truth in software engineering.

Even if further analysis are to be done, the few preliminary results that were partially presented in this contribution tend to demonstrate that a data model and certain patterns are emerging from the vast quantity of data amassed turning the spotlight on a new facet of the empirical reality of software engineering which until today was completely hidden.

5. Acknowledgments

This research would not have been possible without the agreement of the company in which it was conducted, and without the generous participation and patience of the software development team members from whom the data was collected. To all these people, we extend our grateful thanks.

6. References

- [1] Babbie, E., *The Practice of Social Research*, 9th edition, International Thomson Publishing Company, 2001.
- [2] P. D'Astous, and P.N. Robillard, "Empirical Study of Exchange Patterns during Software Peer Review Meetings", *Information and Software Technology*, 44, 2002, pp. 639-648.
- [3] P. D'Astous, P.N. Robillard, F. Détienne and W. Visser, "Quantitative Measurements of the Influence of Participant Roles during Peer Review Meetings", *Empirical Software Engineering*, 6, 2001, pp. 143-159.
- [4] Fenton, N.E. and S.L. Pfleeger, *Software Metrics - A Rigorous & Practical Approach*, PWS Publishing Company, 1997.
- [5] C. Fisher and P. Sanderson, "Exploratory Sequential Data Analysis: Exploring Continuous Observational Data", *Interactions*, 3:2, Mar. 1996, pp. 25-34.
- [6] R.E. Grinter, J.D. Herbsleb, and D.E. Perry, "The geography of coordination: Dealing with distance in R&D work", *GROUP'99: International Conference on Supporting Group Work, Coordination and Negotiation*, 1999, p. 306-315.
- [7] J.D. Herbsleb and R.E. Grinter, "Splitting the Organization and Integrating the Code: Conway's Law Revisited", *Proceedings, International Conference on Software Engineering*, Los Angeles, CA, 1999, pp. 85-95.
- [8] J.D. Herbsleb and D. Moitra, "Guest Editors' introduction: Global software development", *IEEE Software*, 18:2, March/April 2001, pp. 16-20.
- [9] J.D. Herbsleb and A. Mockus, "An empirical study of speed and communication in globally-distributed software development", *IEEE Transactions on Software Engineering*, 29:6, June 2003, pp. 481-494.
- [10] D.L. Jorgensen, *Participant Observation A Methodology for Human Studies*, Applied Social Research Methods Series; v.15, Sage Publications, 1989.
- [11] B. Kitchenham and al., "Preliminary guidelines for empirical research in software engineering", *IEEE Transactions on Software Engineering*, 28:8, 2002, pp. 721-734.
- [12] D.E. Perry, N. Staudenmayer and L.G. Votta, "People, Organizations, and Process Improvement", *IEEE Software*, July 1994.
- [13] P.N. Robillard, "The Role of Knowledge in Software", *Communications of the ACM*, 42:1, 1999, pp. 87-92.
- [14] P.N. Robillard, and M.P. Robillard, "Types of Collaborative Work in Software Engineering", *The Journal of System and Software*, 53, 2000, pp. 219-224.
- [15] P. N. Robillard, P. Kruchten and P. D'Astous, *Software Engineering Process with the UPEDU*, Addison Wesley, Pearson Education, 2002.
- [16] C. Seaman, *Organizational Issues in Software Development: An Empirical Study of Communication*, Ph.D. Thesis, Computer Science Department, University of Maryland, Technical Report CS-TR-3726, UMIACS Technical Report UMIACS-TR-96-94, 1996.
- [17] C. Seaman. "Qualitative methods in empirical studies of software engineering", *IEEE Transactions on Software Engineering*, 25:4, 1999, pp. 557-572.
- [18] W. Tichy, "Should computer scientists experiment more?", *Computer*, 31:5, 1998, pp. 32-40.
- [19] R.J. Walker, L.C. Briand, D. Notkin, C.B. Seaman, W.F. Tichy, Panel: "Empirical Validation-What, Why, When, and How", *Proceedings, International Conference on Software Engineering*, Portland, Oregon, 2003, pp. 721-722.

Annexe J

Empirical Study of *Ad hoc* Collaborative Activities in

Software Engineering

Proceedings of CSAC 2004 - First International Workshop on

Computer Supported Activity Coordination

Porto, Portugal, 2004

Empirical Study of Ad Hoc Collaborative Activities in Software Engineering

Sébastien Cherry¹, Pierre N. Robillard¹

¹ Software Engineering Research Laboratory, Computer Engineering Department,
École Polytechnique de Montréal, C.P. 6079, succ. Centre-Ville, Montréal, Canada
{Sebastien.Cherry, Pierre-N.Robillard} @ polymtl.ca

Abstract. This paper presents empirical research on ad hoc collaborative activities found in an industrial software engineering setting. We believe that a better understanding of these activities and their content will help us to propose software development process enhancements and also provide some insight into the tools needed to support communications in a distributed software development environment. Further details of our motivations are included, followed by a discussion on our research methodology, and, finally, some results of a preliminary analysis confirming the significance of our data and the importance of the observed phenomenon.

1 Introduction

It is well supported in the literature that some problems encountered in software development are not attributable to technical factors, but rather to the human aspects of software engineering [2], [4], [5], [7], [11], [13], [14], [15]. While some aspects, such as “communication” [7], [15], “coordination” [4], [5] and “collaboration” [2], [13], are gaining recognition in the research community, some methodological challenges emerge. Human factors, for example, have been overlooked in the past for many reasons, but principally because of the difficulty in measuring these facets quantitatively [11]. Nevertheless, empirical research in software engineering is growing in popularity and beginning to be adapted to studying this new topic of interest, namely people, and methods and techniques are being borrowed which were formerly used in the human sciences such as psychology and sociology. Like many researchers, we think that this domain will offer research opportunities for years to come.

This paper presents in-progress empirical research in the context of a case study in the industry, and explores collaborative work in software engineering; specifically, the ad hoc collaborative activities that take place during the software development process. By ad hoc collaborative activities, we mean activities which are not formally prescribed, and which occur between two or more developers working on a specific project task and which happen informally and spontaneously. They can take many forms, such as in peer-to-peer conversations, electronic mail exchanges, and so on.

Details of our motivations are to be found in the next section of this paper, followed by a discussion of the research methodology used, including data collection methods, and the need to place greater emphasis on the analysis techniques that will

be used to explore the large quantity of amassed data. Finally, some results of a preliminary analysis are revealed, which confirm both the relevance of the collected data and the importance of the observed phenomenon.

2 Motivations

2.1 Why collaborative work?

As previously mentioned, a growing number of researchers support the view that many of the problems that arise during software development could be imputable to human factors associated with the software engineering process. Perry, Staudenmayer and Votta (1994) [11], among others, believe that too much attention is given to the technological aspects of software engineering. They say that one of the reasons frequently mentioned is the difficulty of measuring these human factors quantitatively. The same comments are also supported by Seaman (1999) [16].

Many approaches have been envisaged to study the human aspects of software engineering. Some researchers have examined the communication occurring during software development [7], [15], while others have studied the coordination aspect [4], [5] and still others were interested in the collaborative work [2], [13].

With regard to collaborative work, Robillard and Robillard (2000) [13] have empirically identified four types of collaborative activities performed during the software engineering process. They defined “ad hoc” collaborative activities, for example, as the work carried out simultaneously by teammates on a particular task of the project and which is not prescribed by a formal process. One of the conclusions of this research was that ad hoc collaborative activities can play a major role in team communication dynamics, accounting for 41% of this dynamics during the case study. Furthermore, these activities constitute the longest of the working sessions, and in addition seem to have an important impact on individual activities, since they often precede long individual working sessions.

Also, Perry, Staudenmayer and Votta (1994) [11] found during another case study that informal communications take up an average of 75 minutes per day per software developer. Seaman (1996) [15] also supports the view that this type of communication is a non-negligible element to be taken into account during a development process, and which is essential if developers are to carry out their tasks adequately.

Because they monopolize quite a considerable part of a software project and constitute an important element of it, as established above, exploratory research is essential to understanding the content of these ad hoc collaborative activities and the communication that ensues, and to measuring the impact, both positive and negative, on the rest of the development process. We believe that such research will help us to subsequently propose software engineering process enhancements which will be better adapted to the human and empirical realities of software development.

By contrast, collaborative software development, also known as “distributed software development”, is an increasingly fashionable domain of research these days.

Both these expressions refer to software development distributed over time and over relatively long distances, something that has become quite common business practice nowadays in cases where it occurs.

However, according to recent research in this domain [5], the distances between the members of virtual teams tends to obstruct informal communications, resulting in problems of coordination. This is another important reason for undertaking research in this field. It will also provide some insight into the tools needed to support communications in a distributed software development environment.

2.2 Why an empirical study?

Empirical research based on the experimental method has been conducted for a long time now, in many of the human sciences, such as psychology and sociology. It is, moreover, very often considered to be the only valid scientific method accepted in these domains. Although empirical research has been conducted in software engineering for many years, it is on a much smaller scale and only quite recently has it seen an increase in popularity. One reason for this is the growing interest in the human aspects of software engineering [16].

Further arguments supporting this new tendency, and strengthening the evidence for it, have been expressed by Tichy (1998) [17] in his paper, “Should Computer Scientists Experiment More?” However, those who uphold this practice in software engineering believe that, since the quantity of empirical research is on the increase, its quality should increase as well [10], [18].

3 Research Protocol

3.1 Problem Statement

Research Objectives. As discussed previously, the importance and the necessity for ad hoc collaborative work and the communications that ensue in software development are widely supported by many authors [2], [4], [5], [6], [7], [11], [13], [15]. Although some research has quantified the importance of the phenomenon, there has been no known attempt to determine and describe the content of that work. These considerations led us to define the following research objectives:

- To observe the collaborative work taking place in a case study in the industry to design a conceptual model and distinguish some patterns of exchanges.
- To characterize the ad hoc collaborative activities found, the communications that ensue, and to identify and describe their content.
- To generate a series of hypotheses which emerges from the results of this research, and which could later be validated by confirmatory research.

Theoretical Relevance of the Research. The fact that there has been little or no empirical study of ad hoc collaborative activities gives this research theoretical relevance. It will make it possible to establish a model of these activities and to gain a sense of the cognitive aspects involved from which we will be able to generate a series of hypotheses to create a theoretical base in this domain.

Practical Relevance of the Research. Based on the fact that good collaboration is an indispensable condition of a software development team working effectively to make a quality product which meets the needs of the user, in the time required and at the expected cost, this research is relevant in practice because it will potentially pave the way to the proposal of improvements to software engineering processes. Also, as previously mentioned, it will allow us to better understand the informal collaboration and communication aspects of software engineering, and provide some insight into the tools needed to support communications in a distributed software development environment.

3.2 Research Methodology

General Approach. The research is carried out by means of participant observation within the framework of a case study in an industrial environment. This type of approach is suitable in our case because this is exploratory research. Also, as Jorgensen (1989) [8] and Babbie (2001) [1] have stressed, study in the field combined with participant observation are appropriate when it is not a question of empirically verifying hypotheses formulated in advance, but rather of inductively generating theories from observations and from the empirical data collected.

Target-setting. The setting in which the chosen software development team works is a large enterprise which produces software for commercial purposes. It is a well-established, mature concern which has been in operation for several years, and where there exists a clearly defined development process. Nevertheless, even if the observations are done in a large company, this last also contains some attributes of smaller organizations since the development of software components is divided into small teams.

Also, based on a common-sense judgment (face validity) [1], we can say that the chosen team of eight individuals is representative of the majority of development teams, with a wide range of ages, amounts of schooling, years of experience in software development and length of service in the company.

Data Collection. The following data collection methods were identified from a preliminary ethnography period within the chosen team which lasted several months.

An initial data collection phase, which took place in the autumn of 2003, is now completed and was spread out over 8 weeks. The results presented in this paper were produced from these earliest data. The purpose of this collection was to gather the

maximum amount of information from the beginning to the end of the development of an update (patch) of a given version of the software produced.

The data collected during this first phase includes:

- 185 hours of audio-video recordings of working sessions over 37 workdays
- The capture of a total of 2496 e-mails exchanged by the 8 teammates
- A daily backup of the source code and other documents and artifacts found

E-mails were captured automatically, by means of triggers defined in the messaging software used in the company. This capture included both e-mails received and those sent by teammates, in order to permit cross-validation.

The daily backup of the source code, and the various documents and artifacts, was available for potential use for subsequent content analysis.

Data Analysis. One of the techniques that will be used for the analysis is Exploratory Sequential Data Analysis (ESDA) [3]. This technique is suited to exploratory research, where the objective is to find answers to research questions or to find patterns among the empirical data and to describe them using, for example, simple statistical representations.

ESDA allows researchers to define, from these descriptions, hypotheses which are subsequently verified by means of confirmatory research using statistical inference methods. However, the important feature of ESDA is that it applies more specifically to research where the sequential integrity of the data must be preserved.

Of the eight operations proposed by ESDA, encoding is certainly the most important. This involves labeling each sequence of data by means of a code formed using a particular syntax and contained in an exhaustive, exclusive and relatively restricted category list, and doing so to decrease the variability of the data, as well as to facilitate its subsequent manipulation. This encoding makes it possible to transform qualitative data into quantitative data, on which it is then possible to perform statistical analysis [3], [16].

The ESDA process, such as proposed by Fisher and Sanderson (1996) [3], is an iterative one, involving the definition of a series of concepts stemming from research questions of interest. The process will subsequently drive what should be observed among the collected raw data and what manipulations should be made to obtain derived data on which it is possible to generate theories or define hypotheses. It is iterative because it is often necessary to revisit certain steps; for example, to add, remove or redefine concepts or categories that are sometimes found intuitively and validated by their statistical representations.

Research Validity. To satisfy the validity criterion for the research, the empirical measure must faithfully translate the empirical reality of the measured phenomenon [1]. To enhance the validity of our research, particular attention is directed to the definition of the concepts or categories chosen to encode the data. These concept definitions, which must arise from the ESDA traditions that concern us [3], as well as from the context of the research field, ensure a degree of representativeness of the phenomenon under study by common-sense validity (face validity) [1], [8].

The concepts or categories under which the data will be encoded, as well as the number of categories chosen, will also be very important as far as content validity is

concerned [1]. This aspect of validity refers rather to the coverage of the meanings encompassed by the concepts. Furthermore, the validity of the connections or the relations (construct validity) [1] should be assured among the concepts forming the theoretical model appearing from the data. This can be done by means of certain correlation measures or statistical associations.

Finally, a data triangulation will be made between qualitative and quantitative data, as well as of data resulting from various sources [16], [18]. Concerning this last point, other phases of data collection are to be anticipated.

4 Preliminary Results

The results presented in this section are the product of a preliminary analysis concerning four of the eight developers in the team who were observed over a period of 8 hours. The choice of these individuals was not made by means of a sampling method, but from direct observations in the field: they had been identified as being likely to work more collaboratively than the others. This choice is justified because the objective of this research is not to find a magic number indicating the time spent on ad hoc collaborative work, but rather to investigate the content of these collaborative activities. It should also be noted that the results below do not take into account e-mail interactions, but only the peer-to-peer conversations and telephone exchanges.

As can be seen from Figure 1, 51% of the time is spent on ad hoc collaborative work, as against 49% for the other types of activities. This result tends to corroborate the observations made in the field that suggested the importance of the phenomenon.



Fig. 1. Distribution of time spent on ad hoc collaborative activities in comparison with other types of activities

Figure 2 indicates the percentage of time spent on ad hoc collaborative activities by the subjects observed. As was noted in the field, subjects MS2 and MS3 seem to have spent a large amount of their time collaborating and communicating in a spontaneous way with their colleagues. This can be explained by the nature of the work performed by these subjects. MS2 occupies the position of project manager in the team, and one of his functions is to circulate relevant information needed by the developers on his team. When we examine more closely the interactions in which MS2 is involved, we note that, for 78.13% of the time, his colleagues initiate the interactions. We can suggest hypothetically that MS2 constitutes a source of the information his colleagues require. However, it was noted in the field that a great deal of the information passed on by MS2 to his teammates is in the form of e-mails. It would be interesting to investigate this method of communication. MS3 is, for his part, respon-

sible for the infrastructure of the software built, and often the individual consulted to resolve problems. He manages several tasks at the same time, which brings him into communication with some of his colleagues more often.

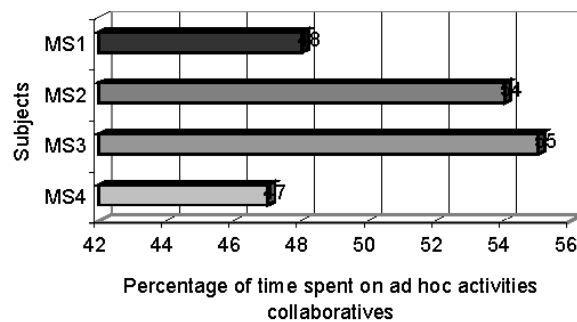


Fig. 2. Percentage of ad hoc collaborative activities per subject

By contrast, the average duration of the interactions analyzed by the four developers is 6:31 minutes, and these interactions involve, on average, 2.3 stakeholders. We should remember that an interaction is defined as a communicative unit which presents an evident internal continuity, while it breaks with what precedes it and what follows it [11]. Moreover, these results were based on a total of 82 observed interactions.

Figures 3 and 4 give an initial outline of a distribution as a percentage with regard to the number of occurrences and time spent on the various categories of ad hoc collaborative activities identified. “Cognitive synchronization” exists when two or more developers exchange information to ensure that they share the same knowledge or the same representation of the object in question. “Problem resolution” occurs when two or more developers are aware of the existence of a problem and attempt by different means to solve the problem or to mitigate it. “Development” occurs when two or more developers contribute to the development of a new feature or component of the software. “Management” is the result of two or more developers coordinating and planning activities such as meetings, common working sessions or setting schedules. “Conflict resolution” is the process of two or several developers taking part in discussions to resolve a difference of opinion. Ad hoc collaborative activities under the “not relevant” category group together all interactions which do not concern the project or the software built.

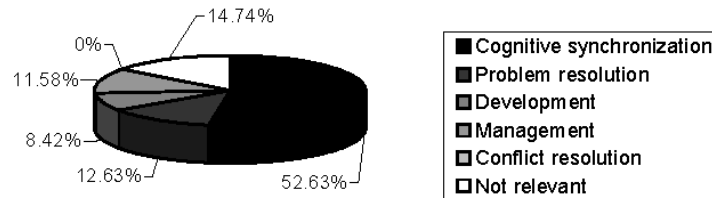


Fig. 3. Distribution in number of occurrences of ad hoc collaborative activities identified

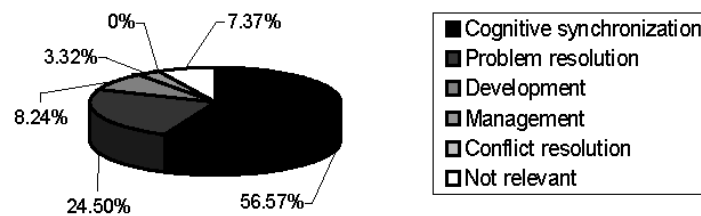


Fig. 4. Distribution in terms of time spent on ad hoc collaborative activities identified

As Figure 3 suggests, 52.63% of the ad hoc collaborative activities that arose are forms of cognitive synchronization. This agrees with the direct observations made in the field. The figure is not surprising when we consider that the exchange of information and knowledge constitutes an essential element in software development, which is to crystallize [16] all the information required in quality software to meet the needs of the user. As shown in Figure 4, cognitive synchronization occupies 56.57% of the time spent on ad hoc collaborative work, which also supports the previous finding.

The other significant category, problem resolution, does not seem important in Figure 3 in terms of number of occurrences. Figure 4, however, suggests that this activity monopolizes almost a quarter of the time spent on ad hoc collaborative work by the four subjects observed. It demonstrates that perhaps, while problem resolution activities are relatively few in number, when they do arise, they monopolize a rather considerable amount of time. An analysis of the mean time spent by sequence as a function of ad hoc collaborative activity tends to confirm this, showing that problem resolution takes up to 9:48 minutes when it occurs, as opposed to the interaction average of 6:31 minutes.

The results relative to management activities also reveal an interesting finding. Unlike problem resolution activities, they are relatively numerous considering the fairly short time that they occupy. This may tend to confirm the theories of certain

authors [7], who maintain that informal communications are necessary in order that the members of a team can coordinate their activities effectively.

5 Conclusion

It is clear, and widely supported, that good collaboration and communication are an essential condition of the successful delivery of a quality product by a software development team, one that meets the user's needs in a timely fashion and at the expected cost.

It was revealed by previous research that ad hoc collaborative activities and informal communications occupy a considerable portion of the time that a developer spends on a software project. However, no research has attempted to describe the content of these activities, which leaves a vast field open for exploration.

The empirical research described in the present paper suggests the importance of investigating this field, because the authors believe that understanding how people collaborate will make it possible to propose practices to enhance collaboration and communication within a development team, as well as improve software development processes.

This article has briefly presented the methodology used to meet our research objectives. It was influenced by previous empirical research which was also aimed at investigating the human aspects of software engineering, but which was, however, adapted to the context of the field on which this study focuses.

The embryonic results that were partially presented in this paper, and that arise from a tiny portion of the considerable quantity of data that was collected, are very interesting. Although more analysis is needed, a model of data and patterns already seems to be emerging which will allow us to subsequently form hypotheses which can be validated by other, confirmatory research, thereby forging a knowledge base in this, as yet unknown, domain of software engineering.

6 Acknowledgments

This research would not have been possible without the agreement of the company in which it was conducted, and without the generous participation and patience of the software development team members from which the data was collected. To all these people, we extend our grateful thanks.

References

1. Babbie, E.: *The Practice of Social Research*, 9th ed., Wadsworth Publishing Company, Belmont, CA (2001)
2. D'Astous, P., Robillard, P.N.: *Les aspects de l'échange d'information dans un processus de génie logiciel*, Rapport interne, École Polytechnique de Montréal, EPM/RT-97-06 (1997)

3. Fisher, C., Sanderson, P.: Exploratory Sequential Data Analysis: Exploring Continuous Observational Data, *Interactions*, Vol.3, No. 2, Mar. (1996)
4. Grinter, R.E., Herbsleb, J.D., Perry, D.E.: The geography of coordination: Dealing with distance in R&D work, *GROUP'99: International Conference on Supporting Group Work, Coordination and Negotiation*, (1999) 306-315
5. Herbsleb, J.D., Grinter, R.E.: Splitting the Organization and Integrating the Code: Conway's Law Revisited, *Proceedings, International Conference on Software Engineering*, Los Angeles, CA (1999) 85-95
6. Herbsleb, J.D., Moitra, D.: Guest Editors' introduction: Global software development. *IEEE Software*, Vol.18 No.2, March/April (2001) 16-20
7. Herbsleb, J.D., Mockus, A.: An empirical study of speed and communication in globally distributed software development, *IEEE Transactions on Software Engineering*, Vol.29, No.6, June (2003) 481-494
8. Jorgensen, D.L.: Participant Observation: A Methodology for Human Studies, *Applied Social Research Methods Series*; v.15, Sage Publications, Newbury Park, CA (1989)
9. Kerbrat-Orrecchioni, C.: *Les Interactions Verbales*, Armand Colin, Paris (1998)
10. Kitchenham B. et al., Preliminary guidelines for empirical research in software engineering, *IEEE Transactions on Software Engineering*, Vol.28, No.8. (2002) 721-734
11. Perry, D.E., Staudenmayer N., Votta, L.G.: People, Organizations, and Process Improvement, *IEEE Software*, July (1994)
12. Robillard, P.N.: Études des aspects cognitifs applicables au génie logiciel, *Rapport interne*, École Polytechnique de Montréal, EPM/RT-96/18 (1996)
13. Robillard, P. N., Robillard, M.P.: Types of Collaborative Work in Software Engineering, *The Journal of System and Software*, No.53 (2000) 219-224
14. Robillard, P. N., Kruchten, P., D'Astous, P.: *Software Engineering Process with the UPEDU*, Addison Wesley, Pearson Education (2002)
15. Seaman, C.: Organizational Issues in Software Development: An Empirical Study of Communication, PhD Thesis, Computer Science Department, University of Maryland, Technical Report CS-TR-3726, UMIACS Technical Report UMIACS-TR-96-94 (1996)
16. Seaman, C.: Qualitative methods in empirical studies of software engineering, *IEEE Transactions on Software Engineering*, Vol.25, No.4 (1999) 557-572
17. Tichy, W.: Should computer scientists experiment more? *Computer*, Vol.31, No.5 (1998) 32-40
18. Walker, R.J., Briand, L.C., Notkin, D., Seaman, C.B., Tichy, W.F.: Panel: Empirical Validation—What, Why, When, and How, *Proceedings, International Conference on Software Engineering*, Portland, Oregon (2003) 721-722